

UNIVERSIDAD AUTÓNOMA DE MADRID
FACULTAD DE PSICOLOGÍA | DEPARTAMENTO DE PSICOLOGÍA BÁSICA

PROGRAMA DE DOCTORADO “COGNICIÓN Y TRASTORNOS”



— Tesis Doctoral —

**UN MODELO DE REPRESENTACIÓN DE
DISTANCIAS MEDIANTE REDES NEURONALES
EN EL CÓRTEX PARIETAL**

FRANCISCO JAVIER SÁINZ ALONSO

DIRECTOR: BERNARDO LLANAS JUÁREZ

TUTOR: JOSÉ MANUEL IGOA GONZÁLEZ

-2012

Agradecimientos

Quiero agradecer sinceramente a todos los que han contribuido de forma directa o indirecta a la elaboración de esta tesis.

En primer lugar, a Bernardo, amigo, tu gran pasión ha sido la investigación, y es tu mejor legado para aquellos que la hemos vivido contigo. Esta tesis no hubiera existido sin tu dirección, y es un tributo a tu memoria. Tu dignidad y tu entusiasmo espero que me acompañen el resto de mi vida.

A José Manuel Igoa, por hacer fáciles los problemas que se plantean con una tesis que se mueve entre tantas aguas y apoyarla.

A los seminarios de la Facultad de Psicología con Angel Rivière, donde aprendí la importancia de formular preguntas. Todos los que estábamos entonces, Fernando Adrover, María Sotillo, Angeles Nogales, Antonieta Cendoya, Marta Morgade, Silvia Español, Adriana Dergam y algunos más, seguiremos recordando aquello.

A mis padres y hermanos, especialmente a mi madre, por haber estado.

Y por último a Rosa, sin tu paciencia y sin tu ánimo esto no hubiera llegado a puerto.

Introducción

El estudio de la mente humana y de su correlato neuronal, el cerebro, ha constituido una fuente inagotable de preguntas e investigaciones a lo largo de la historia. Esas preguntas fueron primero filosóficas, pero luego han sido psicológicas, neurocientíficas, computacionales y todavía hoy no han agotado todos los campos de conocimiento de los que disponemos. Aquellos que hemos creído siempre en la importancia de la pregunta como elemento de conocimiento (y que tanto debemos a Sócrates), estamos más dispuestos a entender el contenido de la pregunta que a plantearnos de qué disciplina viene.

Hay una matriz conceptual respecto a lo mental que debe parte de su origen a un Simposio sobre Teoría de la Información ocurrido en 1956, del 10 al 12 de Septiembre en el Massachusetts Institute of Technology (M.I.T.). En dicho lugar, varios investigadores de disciplinas distintas dialogaron acerca de la comunicación y de las ciencias humanas. Ese interés en unir distintas fuentes de conocimiento en torno a la Cognición ha seguido su camino y a lo largo de todo este tiempo desde 1956 han ido surgiendo distintos puntos de vista con ese denominador común. Se puede hablar de distintas disciplinas que han estado

asociadas al estudio de la Cognición, algunas como la Inteligencia Artificial, han intentado abstraerse del lugar en el que ocurren los procesos cognitivos, otras como la Neurociencia han intentado fijarse sólo en el lugar donde ocurren dichos procesos. Esto habla de la diversidad de enfoques acerca de un mismo problema.

Entre los paradigmas de investigación a que ha dado lugar este impulso científico hay uno que aparece en los años 80 que ha producido un flujo incesante de investigaciones. Se conoce como Conexionismo, y parte del uso de un conjunto de elementos, redes neuronales artificiales, para modelizar la Cognición. Este paradigma está muy vinculado al aprendizaje, ya que las redes neuronales aprenden a partir de ejemplos, construyendo leyes generales. Tiene un componente inductivo muy fuerte, de hecho podemos decir que las redes neuronales son máquinas lógicas de inducción sobre el mundo. Ello plantea muchos problemas: formalizar como se realiza esa inducción, explicar como un problema es reducible para ser resuelto por esas máquinas de inducción, y para aquellos que estamos interesados en el conocimiento humano, si esos mecanismos son asimilables desde el punto de vista biológico a lo que sucede en nuestro cerebro. Este último punto es fundamental, ya que la analogía con las redes neuronales reales que existen en nuestro cerebro constituye uno de los objetivos más ambiciosos del conocimiento científico hoy en día.

Este trabajo que nos ocupa es un proyecto de utilización de redes neuronales MLFN (redes multicapa hacia delante) con el objeto de entender un proceso que ocurre en el córtex cerebral. La Arquitectura y la metodología tradicional de entrenamiento (iterativa) de MLFN plantea muchos problemas como veremos más adelante. Por ello es necesario desarrollar nuevas formas más efectivas de

entrenamiento y nuevos modelos de redes (a las que vamos a llamar Máquinas de Aprendizaje Neuronal).

Nuestro objetivo de partida es que se puede modelizar la representación de las propiedades métricas de los objetos, información que es necesaria para coger dicho objeto, mediante un tipo determinado de red neuronal (NLM), mejorando los resultados de otros tipos de redes en cuanto a garantizar la precisión, el tiempo de ejecución y el tiempo de entrenamiento.

Se define una **máquina de aprendizaje neuronal** (NLM) como cualquier función con parámetros ajustables cuyos componentes elementales sean MLFNs sin procesos de realimentación.

Un ejemplo elemental de NLM es una MLFN. Pero hay otros ejemplos, como las redes neuronales modulares y sus variaciones. En la construcción de una NLM se puede usar un método **directo** o un método **indirecto**. La determinación de los parámetros se puede hacer por **métodos iterativos** o usando **técnicas constructivas**. Las NLMs son un modelo teórico perfectamente adecuado para el aprendizaje (aproximación) de funciones (teoremas de Mhaskar, Hornik, Cybenko). Dada una función concreta a ser aprendida, la determinación de la arquitectura y de los parámetros ajustables de una NLM plantea muchos problemas:

P1. Determinación de la Arquitectura global de la NLM: «¿Cuántas neuronas son necesarias?».

P2. «¿Cómo se seleccionan los ejemplos para que el entrenamiento (aprendizaje) de la NLM sea óptimo?».

P3. «¿Qué algoritmo de aprendizaje es el mejor y más rápido para una arquitectura determinada?».

P4. Correspondencia con procesos biológicos.

Los algoritmos iterativos como la retropropagación (backpropagation) en el caso de MLFNs, presentan los siguientes problemas:

- Determinación poco adecuada de los parámetros ajustables (mínimos locales) lo que lleva a NLMs que aproximan (generalizan) mal.
- Extremada lentitud de entrenamiento.
- Escasa plausibilidad biológica.

En la parte 1 de este trabajo se exponen brevemente los métodos de aprendizaje biológico por medio de neuronas reales y se enuncia el problema de representación métrica de objetos sólidos en el cerebro. Se formula la hipótesis de que esta representación está localizada y constituida por estructuras localizadas en el córtex parietal. Esta hipótesis viene avalada por experimentos y descubrimientos en Neurociencia.

En la parte 2 se define el concepto de MLFN, así como su evolución histórica. Se detallan los teoremas de aproximación que dotan a estas redes de la capacidad de ser aproximadores universales. Se detallan sus generalizaciones (NLM) y los

diferentes métodos para mejorar la determinación de los parámetros ajustables: Técnicas Constructivas, Máquinas de Aprendizaje Extremo y Métodos Indirectos.

En la parte 3 se detalla la metodología a seguir.

En la parte 4 se presenta una técnica constructiva para demostrar el Teorema de Cybenko. Esta técnica está basada en el concepto de interpolación aproximada y constituye una alternativa a la retropropagación para entrenar SLFNs. También se da una demostración constructiva del hecho de que una función afín puede ser aproximada uniformemente por únicamente dos neuronas. Este hecho será de vital importancia en la última parte del trabajo.

La parte 5 presenta un método indirecto para construir NLMs. Este método genera unos árboles neuronales que están basados en un método de aproximación conocido como desdoblamiento adaptativo (adaptive splitting). Este proceso y los árboles neuronales resultantes tienen las siguientes características:

C1. La arquitectura (conexiones, número de neuronas, etc.) es generada automáticamente a partir de la función a aproximar.

C2. Los datos para un aprendizaje óptimo son seleccionados por la misma red.

C3. El aprendizaje es bastante rápido.

C4. Sólo se necesita que las neuronas ejecuten algunas operaciones elementales con las señales que reciben: sumas, producto por escalares, etc. Se supone que la capacidad de tener incorporados procesos de integración aproximada (cubatura) forma parte de la dotación genética de algunas neuronas. De esta manera

se especifica la diferencia entre las características innatas (nativismo) y aquellas obtenidas mediante aprendizaje (empirismo).

La simulación numérica en el caso de la función distancia a un objeto, confirma las propiedades anteriores. En el caso de objetos con forma de poliedro convexo, se llega a la conclusión de que el número de neuronas para aprender (con un nivel de error prefijado) la distancia al poliedro, no depende de su complejidad combinatoria (número de caras y vértices).

Referencias

B. Llanas, F.J. Sáinz, Constructive approximate interpolation by neural networks, J. Comput. Appl. Math. 188 (2006) 283-308.

B. Llanas, F.J. Sáinz, Fast training of neural trees by adaptive splitting based on cubature. Neurocomputing 71 (2008) 3387-3408.

B. Llanas, F.J. Sáinz, Aproximación eficiente de una función afín definida sobre un politopo, por medio de redes neuronales. II Congreso Internacional de Matemáticas en la Ingeniería y la Arquitectura. XV Jornadas ASEPUMA. (2008).

Terminología

AIP	Área Intraparietal
ANN	Red Neuronal Artificial («Artificial Neural Network»)
ASBC	Algoritmo de desdoblamiento adaptativo basado en cubatura
BA	Área de Brodmann
BP	Retropropagación del error («Backpropagation»)
CPP	Córtex Parietal Posterior
ERR	Estimación del Error local de cubatura
ILM	Modelo de Aprendizaje Inicial («Initial Learning Model»)
IT	Córtex Inferotemporal
LO	Complejo Occipital Lateral
LSABF	Algoritmo de Búsqueda local basado en caras
LTD	Depresión a largo plazo («Long Term Depression»)
LTP	Potenciación a largo plazo («Long Term Potentiation»)

MLFN	Red Neuronal Feedforward Multicapa («Multi Layer Feedforward Network»)
MLP	Perceptrón Multicapa («Multi Layer Perceptron»)
MNN	Red Neuronal Modular («Modular Neural Network»)
NLM	Máquina de Aprendizaje Neuronal
NN	Red Neuronal («Neural Network»)
NTASBC	Algoritmo de generación de árboles neuronales por desdoblamiento adaptativo basado en cubatura
PDP	Procesamiento Distribuido en Paralelo
RMNf	Resonancia Magnética Nuclear Funcional
RMSE	Error cuadrático medio («Root Mean Squared Error»)
SLFN	Red Neuronal de una Capa («Single Layer Feedforward Network»)
SPOC	Córtex Parieto-Occipital Superior («Superior Parietal Occipital Cortex»)
STRIRN	Método de regiones de confianza basado en el algoritmo interior-reflexivo de Newton
TEP	Tomografía de Emisión de Positrones
TLFN	Red Neuronal Feedforward de dos Capas («Two Layer Feedforward Network»)
TNLM	Máquina de Aprendizaje Neuronal Entrenada
TNN	Red Neuronal Entrenada («Trained Neural Network»)
XOR	Or-exclusivo («Or-exclusive»)

Índice general

Introducción	II
Referencias	VIII
Terminología	IX
1. Fundamento Biológico	1
1.1. Aprendizaje Biológico	1
1.1.1. Cableado neuronal inicial	3
1.1.2. Modificación de circuitos cerebrales como resultado del aprendizaje	8
1.2. Representación métrica de objetos en el córtex	13

1.2.1.	Dos sistemas visuales	14
1.2.2.	Dos sistemas visuales corticales	16
1.2.3.	Canales múltiples en la vía dorsal	18
1.2.4.	Dos vías visuales en humanos	19
1.2.5.	El córtex parietal humano: Una llamada para la acción . .	20
1.2.6.	¿Por qué debería existir un área especializada en coger objetos?	22
1.2.7.	Un área para coger objetos en monos	24
1.2.8.	Un área para coger objetos en humanos	27
1.2.9.	Nuevas investigaciones sobre un área para coger objetos en humanos	29
2.	El perceptrón multicapa como máquina de aprendizaje	31
2.1.	Definición de Conexionismo	31
2.2.	La arquitectura conexionista genérica: MLFN	36
2.2.1.	Definición de SLFN y MFLN	36
2.2.2.	Propiedades de Aproximación	43

2.3. Panorámica histórica de las MLFN. La Retropropagación como el primer método iterativo para entrenar MLFNs	45
2.4. Inconvenientes de la Retropropagación	48
2.5. Métodos para Construir una Máquina de Aprendizaje Neuronal. .	50
3. El problema de representar objetos en el espacio para la acción:	
Metodología	58
3.1. Introducción	58
3.2. Metodología	60
3.3. Planteamiento del problema	61
3.4. Requisitos de los algoritmos	62
3.5. Formulación matemática y computacional	65
3.6. Entorno experimental	67
4. Un enfoque constructivo de la aproximación de funciones mediante redes neuronales	69
4.1. Introducción	69

4.2.	Interpolación y redes neuronales. El problema de la interpolación exacta con una SLFN	72
4.2.1.	Interpolación y redes neuronales	72
4.2.2.	El problema de la interpolación exacta con una SLFN . . .	78
4.3.	Interpolantes aproximados neuronales unidimensionales	92
4.4.	Aproximación uniforme por medio de interpolantes aproximados .	104
4.5.	Una demostración del Teorema de Cybenko	114
4.6.	Aproximación uniforme de una función afín por medio de dos neuronas	115
5.	Un método indirecto para la construcción de árboles neuronales de aprendizaje rápido. Aplicación a la representación de objetos	124
5.1.	Introducción. Necesidad de Aprendizaje rápido.	124
5.2.	Aproximación por trozos de funciones por el método de división adaptativa basado en cubatura (ASBC).	127
5.3.	Algoritmo para generar el árbol de aproximación	134
5.4.	Estructura del árbol neuronal basado en ASBC	140

5.4.1.	Nodos de cómputo	140
5.4.2.	Nodos de direccionamiento	140
5.4.3.	Estructura del árbol resultante	142
5.4.4.	Computación biológica dirigida por datos	144
5.5.	Comportamiento numérico de NTASBC.	149
5.5.1.	Experimentos en dos dimensiones	150
5.5.2.	Experimentos en tres dimensiones	155
5.5.3.	Número de distancias calculadas por NTASBC	164
5.5.4.	Experimentos de variación de dominio	164
5.6.	Comparación entre redes obtenidas mediante NTASBC y redes SLFN	165
5.6.1.	Comparación entre redes obtenidas mediante NTASBC y redes SLFN entrenadas por un método de regiones de confianza basado en el algoritmo interior-reflexivo de Newton	165
5.6.2.	Tiempos de cálculo individuales	171
5.7.	Conclusiones	172

Conclusión	177
Bibliografía	179
Esquema General de los Programas	197

Índice de figuras

1.1. Distribución lobular del cerebro humano	13
1.2. Vía ventral y vía dorsal. Separación anatómica y funcional	15
1.3. Vista lateral del cerebro de un mono mostrando zonas visuales dorsales y ventrales	18
1.4. Córtex Parietal: (a) cerebro de mono (b) cerebro humano	28
2.1. Arquitectura genérica de red neuronal MLFN	36
2.2. Red neuronal SLFN (Una capa oculta)	39
2.3. Red neuronal TLFN	43
4.1. N_a en el caso de $\sin(2\pi x)$	103
5.1. Generación del árbol de aproximación.	139

5.2. Árbol neuronal generado por NTASBC.	144
5.3. Direccionamiento biológico.	146
5.4. Unidad de retardo.	147
5.5. Unidad inhibitoria.	147
5.6. Figuras usadas en los experimentos 2D.	151
5.7. Resultados de NTASBC en dos dimensiones.	153
5.8. Teselaciones generadas por NTASBC en el caso del hexágono y del polígono no convexo.	154
5.9. Poliedros convexos utilizados en los experimentos 3D.	157
5.10. Poliedro no convexo utilizado en los experimentos 3D.	158
5.11. Resultados de NTASBC en tres dimensiones.	161
5.12. Error de generalización de los árboles generados por NTASBC en tres dimensiones.	162
5.13. Intersección de planos horizontales con las teselaciones generadas por NTASBC.	163
5.14. Experimentos de variación de dominio.	165

5.15. Comparación entre NTASBC y SLFNs entrenadas con STRIRN en el caso de figuras bidimensionales.	168
5.16. Comparación entre NTASBC y SLFNs entrenadas con STRIRN en el caso de figuras tridimensionales.	169
5.17. Crecimiento del tiempo de entrenamiento de SLFNs entrenadas con STRIRN.	170

Índice de Tablas

5.1. Número de triángulos en las teselaciones 2D.	152
5.2. Tiempo de CPU para el entrenamiento (en segundos).	152
5.3. Error de generalización en 2D de los árboles generados por NTASBC.	152
5.4. Número de tetraedros en las teselaciones 3D.	158
5.5. Tiempo de CPU para el entrenamiento (en segundos).	159
5.6. Error de generalización en 3D de los árboles generados por NTASBC.	159
5.7. Número de distancias calculadas por NTASBC.	164
5.8. Resultados de aprendizaje de redes SLFN entrenadas por STRIRN en el caso bidimensional.	167

5.9. Resultados de aprendizaje de redes SLFN entrenadas por STRIRN en el caso tridimensional.	167
5.10. Tiempo de entrenamiento con un número fijo de neuronas (nn=25).	169
5.11. Tiempo de entrenamiento con un número fijo de datos de entrenamiento (ntd=1000).	170
5.12. Tiempo medio de cálculo de distancias individuales (en microse- gundos).	171
5.13. Tiempo medio de cálculo de distancias individuales (en microse- gundos) de los árboles generados por NTASBC.	171

Capítulo 1

Fundamento Biológico

1.1. Aprendizaje Biológico

Al principio del siglo XIX, la célula es reconocida como la unidad fundamental de los organismos vivos. Hasta bien entrado el siglo XX, los neurocientíficos no están de acuerdo en que el tejido nervioso, como el resto de los órganos, está compuesto de células. La razón fundamental de ese primer error es que la primera generación de neurobiólogos tiene dificultad para entender la naturaleza de las células nerviosas con el microscopio y las técnicas disponibles. El problema se agrava debido a las formas extraordinariamente complejas y las ramificaciones de las neuronas individuales. De hecho, hay biólogos que piensan que cada neurona está conectada con sus vecinas por enlaces protoplásmicos, formando una red continua conocida como «reticulum». La teoría reticular de

la comunicación interneuronal tiene a su mejor representante en Camilo Golgi. Esta teoría es reemplazada por lo que se conoce como doctrina de la neurona, representada por Santiago Ramón y Cajal. El trabajo posterior de Sherrington y otros demuestra la transferencia de señales eléctricas entre las neuronas a través de uniones celulares que se llaman sinapsis. Aunque los constituyentes celulares del sistema nervioso humano son de alguna forma similares a los de otros órganos, son inusuales por su número, el cerebro humano tiene 100.000 millones de neuronas. El sistema nervioso tiene un rango más amplio de tipos distintos de célula, en morfología, identidad molecular o actividad fisiológica, que cualquier otro órgano. Durante gran parte del siglo XX, los neurocientíficos han utilizaron el mismo tipo de técnicas desarrolladas por Cajal o Golgi para describir y categorizar la diversidad de los tipos de células del sistema nervioso. Desde finales de los años 70, han aparecido nuevas tecnologías procedentes de los avances en biología molecular y celular, dando a los investigadores herramientas adicionales para estudiar las propiedades de las neuronas. Las neuronas no funcionan de forma aislada, están organizadas en grupos o circuitos neuronales que procesan información específica y son la base de la sensación, percepción y del comportamiento. Las conexiones sinápticas que definen tales circuitos se dan en una red de dendritas, terminales axónicos y células gliales que constituyen en conjunto una región entre cuerpos nerviosos. Aunque la topología de los circuitos neuronales varía de acuerdo a la función a la que sirven, hay algunos hechos característicos. Un factor fundamental es el flujo de información en cualquier circuito neuronal particular, característica esencial para comprender su propósito. Las células nerviosas que llevan información hacia el cerebro o la médula espinal se llaman neuronas aferentes, las que llevan información hacia fuera del cerebro

o la médula espinal se llaman neuronas eferentes. Las interneuronas o neuronas locales de un circuito sólo participan en aspectos locales de dicho circuito. Estas tres clases son los constituyentes básicos de todos los circuitos neuronales.

1.1.1. Cableado neuronal inicial

El establecimiento de los circuitos neuronales iniciales en el sistema nervioso se hace después de que las neuronas ya han sido generadas en el desarrollo y han migrado a sus posiciones finales. Las neuronas en diferentes regiones están conectadas a través de axones. Las conexiones sinápticas se deben hacer con unos patrones pre y postsinápticos adecuados. Los axones en crecimiento tienen la capacidad de navegar milímetros e incluso centímetros para encontrar los patrones sinápticos apropiados. El crecimiento axonal se realiza a través del cono de crecimiento, una estructura especializada del axón que se extiende más allá del cuerpo celular. Los conos de crecimiento son estructuras móviles que exploran el entorno extracelular, determinan la dirección de crecimiento, y guían la extensión del axón en esa dirección. Santiago Ramón y Cajal, ya notó (1890), que cuando los conos de crecimiento se mueven a lo largo de un camino establecido por otros axones, tienden a seguir caminos rectos. En contraste, cuando un axón en crecimiento se extiende a una nueva dirección o alcanza una nueva región donde se debe tomar una elección acerca de que dirección tomar, la estructura del cono de crecimiento sufre cambios rápidos. El cono de crecimiento extiende numerosos «filopodia», sugiriendo una búsqueda activa. Estos cambios de la forma del cono de crecimiento en los puntos de decisión han sido observados tanto en el sistema

nervioso periférico como en el sistema nervioso central. En la periferia, los conos de crecimiento de las neuronas motoras experimentan cambios cuando llegan a los músculos en extremidades inmaduras, presumiblemente facilitando la selección de objetivos apropiados en la musculatura en desarrollo. En el sistema nervioso central, los conos de crecimiento en la espina dorsal, y los nervios ópticos y olfatorios cambian su forma cuando alcanzan los puntos críticos en sus trayectorias. El comportamiento complejo de los conos de crecimiento durante la extensión axonal sugiere la presencia de señales específicas que indican al cono de crecimiento el movimiento en una dirección particular. Además, el cono de crecimiento debe poseer un conjunto especializado de receptores y mecanismos de transducción para responder a estas señales. Otro gran problema es establecer los patrones adecuados de conectividad para atraer los axones a larga distancia, y asegurarse de que los axones no pasan por regiones inapropiadas. Cajal (1913) propuso en el siglo XX que hay señales emitidas en los objetivos que afectan selectivamente a los conos de crecimiento, atrayendo a los axones a los destinos apropiados. Además de la quimioatracción predicha por Cajal, se ha supuesto durante largo tiempo que existe una quimiorepulsión para evitar que los axones terminen en otras regiones. Ha sido en los últimos años que se han empezado a verificar tales hipótesis. Gran parte de la investigación sobre el guiado de axones se ha focalizado en moléculas que orientan a los axones o atraen a las neuronas en crecimiento. Sin embargo, también hace falta decir a las neuronas donde no crecer. A principios de los años 60, Roger Sperry (1963), que más tarde hizo un trabajo pionero sobre la especialización funcional de los hemisferios cerebrales, articuló la hipótesis de la quimioafinidad, basado en el trabajo de los sistemas visuales de ranas y carpas doradas. En estos animales, los terminales de las

células glanglionares de la retina forman un mapa topográfico en el tectum óptico. Cuando Sperry cortó el nervio óptico y permitió que se regenerase, encontró que los axones retinales reestablecían el mismo patrón de conexiones en el tectum. Incluso si se rotaba el 180 grados, los axones se regeneraban a sus destinos originales tectales. Por ello Sperry propuso que cada célula tectal lleva una etiqueta de identificación, supuso además que los terminales en crecimiento de las células retinales ganglionares tienen etiquetas complementarias, de tal forma que tienen una localización específica en el tectum. En terminología moderna, estas etiquetas químicas son moléculas de adhesión o reconocimiento, y la afinidad que engendran es una combinación selectiva de moléculas receptoras sobre los conos de crecimiento y moléculas sobre las células tectales que señalan sus posiciones relativas. Las interacciones positivas son debidas probablemente a la adhesión incrementada de los conos de crecimiento al sustrato, mientras el problema para crecer dentro de regiones inapropiadas puede resultar de interacciones repulsivas que tienden al colapso de los conos de crecimiento. Después de alcanzar el objetivo correcto o la región correcta, los axones deben determinar localmente que células innervar entre una variedad potencial de sinapsis locales. La elección disponible para un axón incluye establecer contactos sinápticos, retraerse y dirigirse a otro objetivo, o decaer en el establecimiento de conexiones estables. Debido a la complejidad de los circuitos cerebrales, este tema ha sido estudiado fundamentalmente en el sistema nervioso periférico, particularmente en la innervación de las fibras musculares y de las células ganglionares autonómas de las neuronas espinales motoras. La especificidad sináptica fue estudiada por primera vez por el neurofisiólogo inglés John Langley (1895) al final del siglo XIX. Cuando se produce la sinaptogénesis, las neuronas y sus objetivos, tanto en

el sistema nervioso central como en el sistema nervioso periférico, se asocian de acuerdo a un sistema de preferencias continuamente variable. Las células objetivo residentes en músculos, ganglios del sistema nervioso autónomo, o en cualquier otro sitio, no son exclusivas respecto a la inervación que pueden recibir. Esta relativa promiscuidad puede causar problemas después de un daño neuronal, dado que los patrones de regeneración de la inervación periférica no son siempre apropiados. Distintas observaciones muestran que gran parte de las mismas moléculas de adhesión que participan en el guiado axonal contribuyen a la identificación y estabilización de una localización sináptica sobre las células objetivo, así como la capacidad de un axón en crecimiento para reconocer localizaciones específicas como óptimas. La formación de los contactos sinápticos entre axones en crecimiento y sus células objetivo marca el principio de un nuevo estadio del desarrollo. Una vez que los contactos sinápticos se han establecido, las neuronas se vuelven dependientes en algún grado de la presencia de sus células objetivo para la supervivencia continua y la diferenciación. En la ausencia de sinápsis objetivo, los axones y dendritas de neuronas en desarrollo se atrofian y las células nerviosas inervantes pueden morir. Esta dependencia a largo plazo entre neuronas y sus células objetivo se conoce como interacción trófica. Una cuestión que se plantea es la dependencia tan fuerte de las neuronas respecto de las células objetivo. La respuesta descansa en la escala cambiante del sistema nervioso en desarrollo y al cuerpo al que sirve, y la necesidad de relacionar el número de neuronas de una población determinada con el tamaño de sus objetivo. Una estrategia general en el desarrollo de los vertebrados es la producción de un exceso de células nerviosas, la población final se establece por la muerte de aquellas neuronas que fallan en la interacción con sus objetivos.

Una vez que las poblaciones neuronales se han establecido, las interacciones tróficas continúan modulando la formación de conexiones sinápticas, empezando en la vida embrionaria y extendiéndose más allá del nacimiento. Entre los problemas que deben ser resueltos durante el establecimiento de la inervación está asegurar que cada célula objetivo está inervada por el número correcto de axones, y que cada axón inerva el número correcto de células objetivo. Ajustar correctamente esa correspondencia es uno de los grandes objetivos de las interacciones tróficas entre neuronas en desarrollo y células objetivo, y es necesario para establecer los circuitos apropiados para las demandas funcionales de cada organismo individual.

El patrón de conexiones sinápticas que emerge en el adulto no es una consecuencia simple de las identidades bioquímicas de los patrones sinápticos en el desarrollo. El cableado del adulto es el resultado de un proceso flexible en el que las conexiones neuronales se forman, se quitan y se remodelan de acuerdo a circunstancias locales que reflejan restricciones moleculares y la actividad eléctrica. Estas interacciones garantizan que toda célula objetivo se inerve por el número correcto de entradas y sinapsis, y que todo axón inervante contacte con el número correcto de células objetivo con el número apropiado de terminales sinápticos. Así, la regulación de la convergencia (el número de entradas a una célula objetivo) y la divergencia (el número de conexiones hechas por una neurona) en el sistema nervioso en desarrollo es una consecuencia clave de las interacciones tróficas entre neuronas y sus células objetivo. La regulación de la convergencia y la divergencia por interacciones neurotróficas está influenciada por la forma de las neuronas, particularmente la elaboración de dendritas, un hecho que está sujeto al control neurotrófico. Las interacciones tróficas regulan tres

pasos esenciales en la formación de los circuitos neuronales maduros, primero, haciendo corresponder el número de aferencias a los objetivos disponibles, segundo, regulando el grado de invervación de las aferencias individuales y las células postsinápticas, y tercero modulando el crecimiento y forma del axón y las ramas dendríticas.

1.1.2. Modificación de circuitos cerebrales como resultado del aprendizaje

Una vez que las conexiones cerebrales se han establecido, los patrones de actividad neuronal (incluyendo los que son elicitados por la experiencia) modifican las sinapsis. La actividad neuronal generada por las interacciones con el mundo exterior en la vida postnatal son el mecanismo por el cual el entorno puede influir sobre los circuitos sinápticos del cerebro en desarrollo. La actividad neuronal generada por las interacciones con el mundo en la vida postnatal proporciona un mecanismo por el cual el entorno puede influir en la estructura y función del cerebro. Esta influencia de la actividad sobre el cerebro en desarrollo se manifiesta durante ventanas temporales conocidas como periodos críticos. En algunos casos, como la adquisición del lenguaje, las influencias instructivas del ambiente son requeridas para el desarrollo normal del comportamiento. Algunos comportamientos sólo se expresan si los animales tienen ciertas experiencias específicas durante un periodo restringido postnatal. Por otro lado, los periodos críticos sensoriales o motores, o para comportamientos complejos como el lenguaje no están bien delimitados. En 1949, D.O. Hebb hipotetizó que la

actividad coordinada de un terminal presináptico y una neurona postsináptica refuerza la conexión entre ellas. El postulado de Hebb, como se le conoce, fue formulado originalmente para explicar la base celular del aprendizaje y la memoria, pero este concepto general se ha aplicado a situaciones que involucran modificaciones a largo plazo en la fuerza sináptica, incluyendo aquellas que ocurren durante el desarrollo de los circuitos neuronales. La capacidad del sistema nervioso para cambiar (también llamada plasticidad neuronal) es obvia durante el desarrollo de los circuitos neuronales. El cerebro adulto también posee plasticidad neuronal para aprender tareas nuevas, recordar y responder a posibles traumas a lo largo de la vida. Aunque los mecanismos responsables de los cambios en el cerebro adulto no son conocidos completamente, las funciones neuronales alteradas en la madurez descansan en cambios en la fuerza de las sinapsis existentes. Los cambios en los circuitos neuronales también pueden ocurrir por formación de nuevos terminales axónicos y dendritas. Hay cambios muy radicales que ocurren cuando el sistema nervioso central es dañado por un trauma o enfermedad, aunque la regeneración de conexiones en el cerebro y la médula espinal está limitado. Un obstáculo obvio para explorar el cambio en el cerebro de humanos y otros mamíferos es el número de neuronas y la complejidad de las conexiones sinápticas. Es muy difícil atribuir una modificación en el comportamiento a cambios en las propiedades de neuronas específicas o sinapsis. Una forma de evitar este dilema es examinar la plasticidad en sistemas nerviosos simples, y asumir que se mantienen esas propiedades en organismos superiores. Uno de los ejemplos más conocidos de esta vía de investigación son los trabajos de Erich Kandel (Blankenship, Wachtel y Kandel, 1971). usando el molusco marino «*Aplysia californica*». La «*Aplysia*» exhibe formas elementales

de plasticidad comportamental. Una forma es la habituación, un proceso que causa que el animal responde menos a las ocurrencias repetidas de un estímulo. Otra forma de plasticidad es la sensibilización, un proceso que permite que un animal generalice las respuestas aversivas a un estímulo nocivo. La actividad sináptica se modifica durante la habituación y la sensibilización. Durante la habituación, la transmisión en la sinapsis entre las neuronas sensoriales y motoras se deprime. Este debilitamiento de la transmisión sináptica se conoce como depresión sináptica, y se ha demostrado que causa una reducción en el número de vesículas sinápticas. En contraste, la sensibilización añade neuronas adicionales. Estos estudios de «Aplysia», y otros trabajos como los de la mosca de la fruta, han llevado a los investigadores a generalizar sobre los mecanismos neuronales en el sistema nervioso adulto de vertebrados y mamíferos. Primero, la plasticidad del comportamiento puede venir claramente de los cambios en la eficacia de la transmisión sináptica. Segundo, estos cambios en la función sináptica pueden ser efectos a corto plazo que producen cambios en las proteínas sinápticas, o cambios a largo plazo que requieren cambios en la expresión genética, nueva síntesis de proteínas e incluso crecimiento de nuevas sinapsis o eliminación de las existentes. La evidencia para la plasticidad sináptica en el sistema nervioso de los mamíferos es amplia. Los mecanismos de la plasticidad sináptica ocurren en escalas de tiempo desde milisegundos a días, semanas o más allá. Las formas de plasticidad a corto plazo han sido muy estudiadas en gran detalle en las sinapsis neuromusculares periféricas. La activación repetida de la unión neuromuscular produce varios cambios que varían en dirección y duración. La facilitación sináptica es un incremento transitorio en la fuerza de la sinapsis, ocurre cuando dos o más potenciales de acción invaden el terminal presináptico

en una sucesión rápida. La transmisión sináptica puede disminuir después de una actividad repetida, recibiendo el nombre de depresión sináptica. Ocurre debido a que el conjunto de vesículas sinápticas disponibles para la fusión es menor. Durante ese periodo, la fuerza de la sinapsis declina hasta que se vuelve a reciclar la cantidad de neurotransmisor en las vesículas. Fenómenos como la facilitación o la depresión pueden modificar la transmisión sináptica brevemente, pero no pueden ser responsables de los procesos de memoria o de plasticidad que persisten en el tiempo. Las formas de plasticidad a largo plazo se consideran relacionadas con el aprendizaje y la memoria. Algunos patrones de actividad sináptica en el sistema nervioso central producen un incremento en la fuerza sináptica conocida como potenciación a largo plazo (LTP), mientras otros patrones de actividad producen una depresión a largo plazo (LTD). Las siglas en inglés expresan sólo la dirección de la eficacia sináptica, ya que los mecanismos moleculares y celulares involucrados son variados. La potenciación a largo plazo ha sido estudiada en las sinapsis excitatorias del hipocampo de los mamíferos, un área del cerebro relacionada con la memoria. En humanos, la imagen funcional que muestra el hipocampo humano se activa durante ciertas tareas de memoria, y el daño del hipocampo produce una incapacidad para formar ciertos tipos de recuerdos. En ratas, las neuronas del hipocampo disparan potenciales de acción sólo cuando el animal está en ciertos lugares. Esas neuronas de lugar codifican memorias espaciales, de hecho si las ratas tienen dañado el hipocampo, tienen problemas para tareas espaciales. El trabajo sobre LTP comienza a finales de los 60, cuando Timothy Bliss y Terje Lomo (1973), trabajando en el laboratorio de Per Andersen en Oslo, Noruega, descubrieron que unos pocos segundos de estimulación eléctrica de alta frecuencia puede reforzar la transmisión sináptica

del conejo durante días o incluso semanas. Aunque originalmente se pensaba que sólo ocurría en la formación del hipocampo, también ocurre en el cerebelo, regiones neocorticales como la amígdala, el sistema nervioso periférico, la unión neuromuscular y la sinapsis sensoriomotora de la «Aplysia». No existe un mecanismo universal para inducir LTP, dependiendo de las condiciones experimentales se pueden usar distintos mecanismos para la misma sinapsis. Si las sinapsis continúan incrementando su fuerza como resultado de LTP, alcanzarían un nivel de eficacia máxima, haciendo difícil almacenar nueva información. Así, lo lógico es que existan otros procesos que debiliten selectivamente conjuntos de sinapsis. Este proceso es la depresión a largo plazo o LTD. LTD parece que es el mecanismo por el que el aprendizaje se codifica en el cerebelo (M. Ito, 2001), así como el proceso por el que LTP puede ser revertido en el hipocampo y el neocortex. La forma del LTD puede variar en diferentes regiones cerebrales y a veces entre diferentes entradas a la misma región cerebral. En resumen, la plasticidad del comportamiento requiere cambios dependientes de la actividad en las sinapsis que conduce a cambios en las conexiones funcionales dentro y entre los circuitos neuronales. Estos cambios en la eficacia y en la geometría local de la conectividad proporcionan una base no sólo para el aprendizaje, la memoria y otras formas de plasticidad, sino también para algunas patologías.

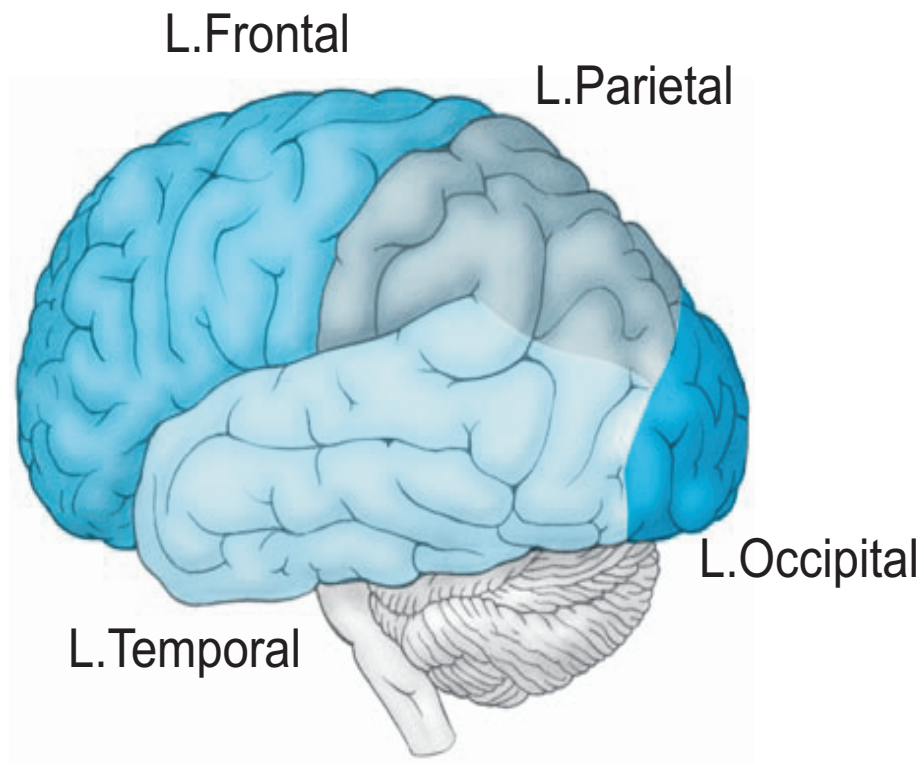


Figura 1.1: Distribución lobular del cerebro humano

1.2. Representación métrica de objetos en el córtex

La separación funcional y anatómica entre las dos vías de procesamiento en el córtex en los primates, llamadas vía ventral y vía dorsal ha sido una hipótesis de trabajo que se ha ido consolidando a lo largo del siglo XX y principios del XXI. En el caso del córtex humano ha sido gracias a los estudios de neuroimagen que se ha podido confirmar dicha separación. Esas dos vías de procesamiento no han recibido igual número de investigaciones, siendo mayor el número de las que se han dedicado a la vía ventral.

La metodología que se ha utilizado en la mayor parte de los estudios ha consistido en estudios anatómicos y registros de actividad cerebral de distintos tipos, siendo decisiva la aportación de técnicas de neuroimagen como la Tomografía por Emisión de Positrones (TEP) y la Resonancia Magnética Nuclear Funcional (RMNf). Estas técnicas, que se utilizan a la vez que los sujetos experimentales realizan determinadas acciones, han conseguido hacer avanzar la comprensión de los procesos de identificación y localización de objetos en el espacio. En un gran porcentaje, las investigaciones se han realizado con primates superiores, porque sólo muy recientemente los humanos han entrado en dicha experimentación.

1.2.1. Dos sistemas visuales

El concepto de dos sistemas visuales aparece en un artículo de Schneider (1969), que propone una separación anatómica de dos sistemas visuales, basándose en estudios sobre el hamster. Considera una vía procedente de la retina a través del núcleo geniculado lateral que va hasta el córtex visual primario (V1) y es responsable de la discriminación de patrones, mientras la entrada desde la retina al tectum (colículo superior en mamíferos) genera respuestas de orientación. Esta distinción anatómica y funcional ya apunta hacia la dualidad conocida de sistemas contestando respectivamente «¿Qué es?» y «¿Dónde está?».

Esa hipótesis de dos sistemas visuales se ve reforzada posteriormente por experimentos con primates. A partir de estudios acerca del comportamiento visomotor en monos con el cerebro dividido, Trevarthen (1968) concluye que

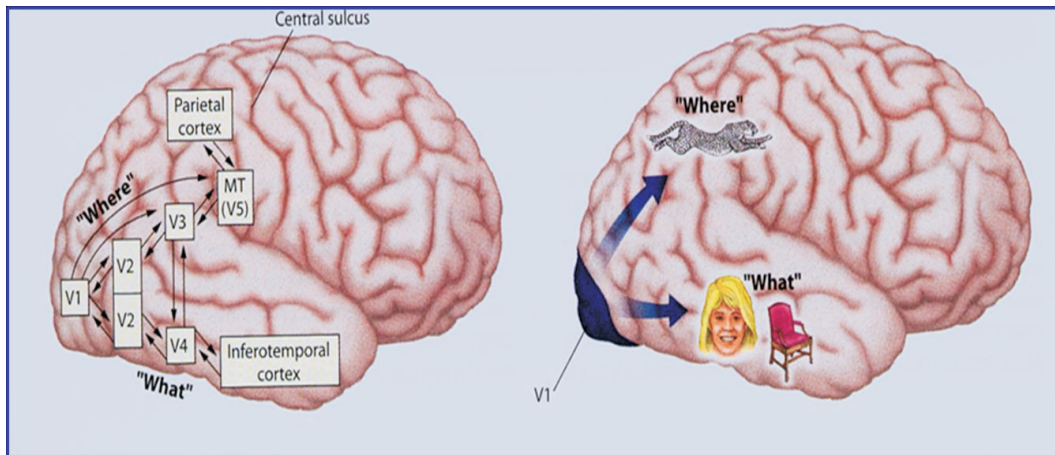


Figura 1.2: Vía ventral y vía dorsal. Separación anatómica y funcional

el sistema visual subcortical sirve a la visión ambiente o periférica, mientras el sistema visual cortical sirve a la visión focal o foveal. El propone que la visión periférica coordina la interacción entre el sujeto y el mundo externo durante la locomoción, mientras la visión focal se usa para la identificación de objetos. Experimentos como el de Humphrey y Weiskrantz (1967), muestran la existencia de funciones visuales residuales en monos con grandes lesiones del córtex estriado, que son capaces de generar respuestas motoras tales como alcanzar objetos, aunque su visión ordinaria está dañada seriamente. Todas estas observaciones señalan el papel de las áreas subcorticales en la visión y su relación con los movimientos del brazo y del cuerpo.

1.2.2. Dos sistemas visuales corticales

Ha sido recientemente, cuando el énfasis de la investigación en primates ha conducido a la separación de dos vías visuales de procesamiento para la visión espacial y para el reconocimiento de objetos dentro del córtex cerebral, relegando el papel de la vía subcortical. Basándose en estudios con lesiones y en experimentos motores con monos, Ungerleider y Mishkin (1982), proponen dos vías procedentes de V1 que se dividen en una vía ventral proyectándose al córtex inferotemporal y una vía dorsal con su nodo central en el córtex parietal posterior. En su propuesta, la vía ventral está dedicada a la identificación de objetos, codificando hechos tales como forma y color, mientras la vía dorsal está dedicada a la codificación espacial de estímulos visuales. La vía dorsal corresponde a una transmisión de información hacia arriba desde el lóbulo occipital hasta el lóbulo parietal y la vía ventral corresponde a una transmisión hacia abajo desde el lóbulo occipital hasta el lóbulo temporal.

En los años siguientes, la idea sobre dos sistemas corticales visuales ha sido elaborada varias veces. En el artículo de 1982, Ungerleider y Mishkin se fijan en las funciones perceptuales, pero estudios posteriores que incluyen comportamiento motor han mostrado que la vía dorsal tiene un papel principal en la coordinación de las transformaciones visomotoras. Por ejemplo, monos con lesiones en el lóbulo parietal posterior tienen dificultades para alcanzar o coger comida con la mano en el lado del cuerpo opuesto a la lesión (contralesional) (Faugier-Grimaud, Francois y Stein, 1978). Además, experimentos con neuronas individuales (Taira, 1990) han revelado que las neuronas en el área de Brodmann

(BA) 7a en el córtex parietal se activan con ciertas formas de objetos o configuraciones de la mano. Utilizando estos resultados, Goodale y Milner (1995) sugieren que la vía dorsal está involucrada en el análisis espacial que es necesario para componer comandos motores necesarios para las acciones visualmente guiadas. Estas acciones son movimientos dirigidos a objetos cuya identificación perceptual es analizada en la vía ventral.

Hay al menos dos buenas razones para atribuir el control visomotor a la vía dorsal. Primero, los tipos de información procesados en este sistema, tal como el movimiento, orientación y la distancia al objeto, están más relacionados con comandos motores que con parámetros estáticos como forma o color; segundo, varios experimentos y casos clínicos han mostrado que el comportamiento visomotor está específicamente afectado por lesiones en áreas parietales y no por daño en áreas temporales inferiores. Así el concepto de vías paralelas en el córtex parietal posterior en organizar la acción orientada a los objetos, complementa el papel de las áreas occipito-temporales en la identificación de objetos.

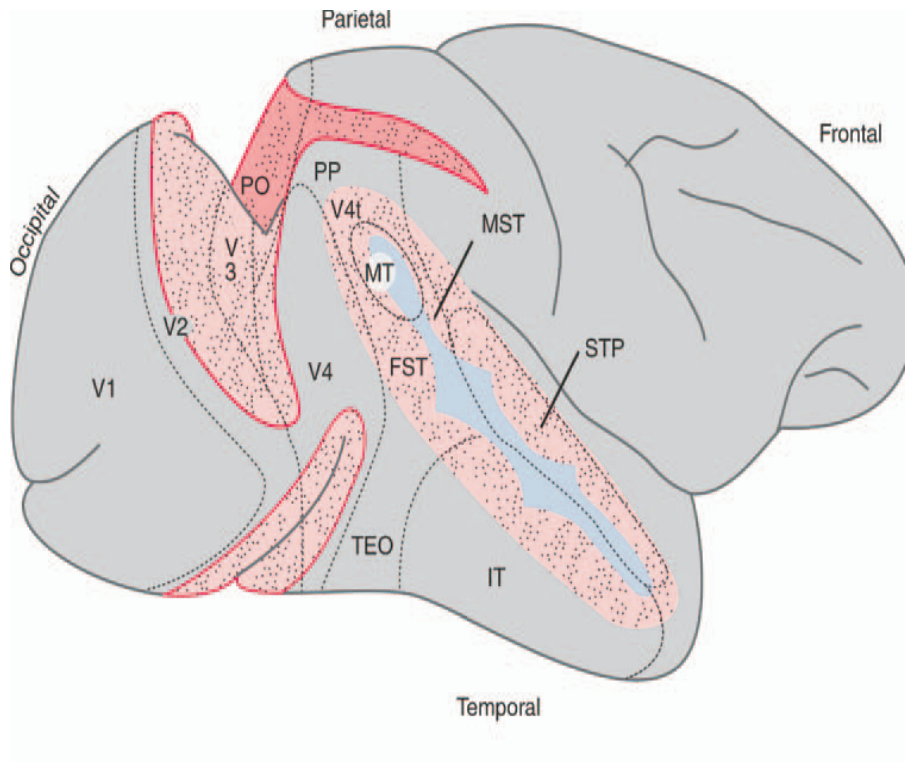


Figura 1.3: Vista lateral del cerebro de un mono mostrando zonas visuales dorsales y ventrales

1.2.3. Canales múltiples en la vía dorsal

La vía dorsal puede ser dividida en al menos dos subsistemas anatómicos o canales, uno para alcanzar y otro para coger. El primero, el cual se proyecta a la parte dorsal del córtex premotor, codifica la dirección de los movimientos, tales como alcanzar un objeto, e incluye mecanismos para computar la distancia y representar la localización. El otro está especializado en la forma de la mano para coger objetos y requiere información acerca de las propiedades intrínsecas del objeto, tales como forma y tamaño.

Esta disociación se demuestra en monos por la inactivación de una región limitada del córtex parietal, el área anterior intraparietal (AIP) (Gallese, Murata, Kaseda, Niki y Sakata, 1994), la cual no produce déficit en transportar la mano a un cierto punto del espacio pero causa problemas graves en la fase de prensión, especialmente en movimientos finos. El déficit es visomotor y no puramente motor porque se corrige bajo guía táctil: tan pronto como el mono toca incorrectamente el objetivo, la mano se ajusta a la configuración apropiada.

1.2.4. Dos vías visuales en humanos

El paradigma de dos sistemas visuales corticales en su versión simple espacio versus objeto ha sido estudiada en humanos usando trabajos perceptuales puros, mientras se monitoriza la actividad cerebral con TEP (Tomografía de emisión de positrones) o con otros métodos de neuroimagen en sujetos normales. Cuando el trabajo implicaba reconocimiento de patrones visuales, la activación cortical ocurría en el área occipito-temporal, es decir, en la vía ventral, mientras en tareas de «matching» espaciales, se activaban las áreas corticales localizadas en el lóbulo parietal superior (Haxby, Grady, Horwitz, Ungerleider, Mishkin, Carson, Herscovitch, Shapiro y Rapoport, 1991).

Además, la división de vía dorsal y ventral está apoyada por amplios estudios sobre alcanzar y coger objetos en sujetos humanos con lesiones en los lóbulos parietales (Jeannerod, 1988, 1995). Los pacientes con lesiones parietales que producen ataxia óptica fallan en alcanzar objetos con la mano contralateral a la lesión y fallan también en la forma de la mano antes de coger un objeto y a veces

cometen errores en la orientación de la mano (Jeannerod, 1994). En contraste con esto, las lesiones en la vía ventral producen déficits en el reconocimiento, pero no afectan al coger o al orientar la mano en el espacio (Goodale, Milner, Jakobson y Carey, 1991).

Se puede hacer una formalización de estos hechos. Un objeto puede ser analizado en términos de dos sistemas de coordenadas: las propiedades extrínsecas relativas a la posición espacial son registradas en coordenadas centradas en el cuerpo y usadas para generar movimientos para alcanzar objetos; las propiedades intrínsecas de los objetos, tales como la forma o la textura, están codificadas en coordenadas centradas en el objeto. El procesamiento de propiedades intrínsecas se divide en una representación semántica para la identificación de objetos y el reconocimiento, que tiene lugar en la vía ventral, y una transformación visomotora, que está localizada en la vía dorsal.

1.2.5. El córtex parietal humano: Una llamada para la acción

Los estudios de imagen cerebral humana son fundamentales hoy en día para la comprensión de la visión y de la cognición visual. Los estudios más importantes se han focalizado en áreas específicas – en jerga de neuroimagen, «regiones de interés» (ROI)– más que en producir listas de regiones activadas por un estímulo particular o tarea o empleando técnicas meta-analíticas, lo cual puede ser desaconsejable en zonas que contienen distintas subregiones.

Lo típico es la identificación de áreas nuevas usando criterios de otras especies o de entornos de procesamiento cognitivo sugeridos por técnicas conductuales o neuropsicológicas. A pesar del «mapeo» exitoso de áreas en las áreas visuales tempranas del córtex occipital y de áreas visuales perceptuales del córtex temporal, hay una escasez sorprendente de exploraciones sistemáticas en el córtex parietal posterior (CPP, las regiones de los lóbulos parietales posteriores al córtex somatosensorial). El CPP forma parte de la «vía dorsal», que se conjetura usa información visual para guiar las acciones, en contraste con la «vía ventral», que hipotéticamente usa información visual para tareas perceptuales tales como el reconocimiento de objetos. Mientras algunas funciones de la vía dorsal, tales como la percepción de movimiento, han sido bien estudiadas, se conoce poco acerca de las áreas y del funcionamiento del CPP humano. Esto es probablemente debido, en parte, a la facilidad relativa de estudiar tareas perceptuales comparado con las tareas de acción en un entorno de neuroimagen. En particular, las funciones parietales tales como la atención, memoria de trabajo, planificación de movimiento del ojo y codificación espacial son componentes comunes de muchas tareas cognitivas y pueden encontrarse en gran parte de la activación que se encuentra a lo largo de muchas tareas funcionalmente distintas. Como han sugerido Culham y Kanwisher (2001), en algunos casos, las demostraciones de lo que no activa el córtex parietal pueden ser más sorprendentes e informativas que las demostraciones de lo que sí activa. Sin embargo, esta aproximación requiere un examen de la actividad regional a lo largo de numerosas tareas y condiciones de control para determinar realmente su papel en la cognición.

A pesar de estas dificultades, con estudios de imagen cuidadosos, puede ser posible identificar áreas funcionales dentro del córtex parietal. La evidencia

neurofisiológica de primates no humanos ha sugerido un mosaico de sub-regiones especializadas dentro del córtex parietal. Determinadas áreas del córtex del macaco parecen específicas para funciones tales como movimientos del ojo y atención, alcanzar objetos, coger objetos, procesamiento de forma y orientación de superficies, y movimiento cerca de la cabeza (Andersen, Snyder, Bradley y Xing, 1997). La revisión de Culham y Kanwisher (2001) de la neuroimagen del córtex parietal humano sugería que áreas similares pueden existir en el córtex parietal humano, en un orden similar neuroanatómico al del mono.

Hay investigaciones recientes que ilustran cómo la imagen cerebral, guiada por el conocimiento de la fisiología del mono, puede explorar el cerebro humano buscando posibles homologías. Los resultados se han focalizado en una región en el sulcus intraparietal anterior que parece estar especializada en la tarea de coger objetos, tanto en monos como en humanos.

1.2.6. ¿Por qué debería existir un área especializada en coger objetos?

Se puede considerar el acto de abrir una puerta. Los humanos reconocen fácilmente los mecanismos comunes para abrirla, tales como un pomo, un tirador o una palanca. Una vez que se identifica el tipo de mecanismo, por ejemplo un tirador vertical, la mano debe ser transportada al tirador, la mano debe estar orientada con una apertura vertical entre los dedos y el pulgar, y la mano abierta debe ser lo suficientemente grande para rodear el tirador antes de agarrarlo. Ambas tareas, el reconocimiento del tirador de la puerta y la colocación de la

mano para coger objetos requiere procesamiento de objetos para determinar las características del tirador. Sin embargo, los requisitos computacionales de estas dos funciones son considerablemente diferentes. En el caso del reconocimiento, el sistema debe ser capaz de identificar el tirador de la puerta desde una variedad de localizaciones potenciales, distancias y puntos de vista. Sin embargo, cuando la persona va a coger el tirador, el punto de vista y la información de la distancia son críticos para codificar con precisión la localización espacial del tirador relativo al brazo, mano o dedos.

De hecho, los requisitos computacionales de los mecanismos perceptuales y de acción parecen suficientemente diferentes para ser desarrollados en distintos sistemas del cerebro. Por ejemplo, un paciente, D.F., con agnosia visual para la forma y pobres capacidades de reconocimiento puede sin embargo coger un objeto de forma precisa (Goodale, Milner, Jakobson y Carey, 1991). Por contra, un paciente con ataxia óptica pueden fácilmente reconocer objetos pero falla para conformar la forma de la mano apropiadamente para coger objetos (Jakobson, Archibald, Carey y Goodale, 1991). Basándose en tales datos, así como en la evidencia conductual de sujetos normales, se ha sugerido que el reconocimiento de objetos se ejecuta en la vía ventral desde el córtex occipital al temporal mientras la prensión de objetos se desarrolla en la vía dorsal desde el córtex occipital al córtex parietal (Goodale y Milner, 1992; Milner y Goodale, 1995).

Los estudios de neuroimagen han tenido éxito en identificar los sustratos de la vía ventral para la percepción de objetos. Un área clave que está involucrada en la percepción de objetos y en el reconocimiento es el complejo occipital lateral (LO) en la vía ventral (Grill-Spector, Kourtzi y Kanwisher, 2001). Como parece

lógico, el área LO es invariante al tamaño y la posición, aunque no al punto de vista. La homología entre LO y las áreas del mono involucradas en la percepción de objetos no está todavía clara, aunque el córtex inferotemporal del macaco (IT) es probablemente homólogo.

La caracterización de los sustratos de la vía dorsal involucrados en el procesamiento de objetos ha sido mucho más limitada. La fisiología del mono sugiere que un área del córtex parietal está especializada en coger objetos y otras formas de manipulación de la mano que requieren procesamiento de características geométricas de los objetos. Basándose en estos resultados del mono, se han usado técnicas de imagen cerebral funcional para investigar si un área similar existe en el hombre y como sus propiedades difieren de las áreas para objetos dentro de la vía ventral.

1.2.7. Un área para coger objetos en monos

El estudio inicial del córtex parietal posterior constituyó un desafío para los fisiólogos porque, de forma distinta a las áreas básicas sensoriales y motoras, las neuronas del CPP eran muy complejas. Las neuronas del CPP tienen combinaciones de respuestas visuales, motoras, somatosensoriales, de movimiento y de movimiento del ojo. Dos poblaciones particularmente intrigantes eran las neuronas de «proyección del brazo» y de «manipulación de la mano» que se encontraban en el banco posterior del sulcus intraparietal anterior (dentro del área 7 de Brodmann). Si las neuronas de proyección del brazo respondían cuando el mono alcanzaba con el brazo un objetivo, las neuronas de manipulación

de la mano se disparaban cuando el mono manipulaba un objeto con la mano, como en una acción para coger objetos. En ambos casos, la activación sólo ocurría cuando el mono estaba intentando activamente obtener un objeto o llegar a una localización, pero no cuando los brazos o las manos eran estimulados pasivamente. Mountcastle (1975) informaba que las neuronas de manipulación de la mano se disparaban furiosamente cuando el mono manipulaba la mano dentro de una caja pequeña para conseguir comida («pestañeo» como el lo llamaba).

Las neuronas de manipulación de la mano han sido estudiadas sistemáticamente por Sakata y sus colegas (1995). Ellos informaban de que las neuronas en un área a la que llamaban AIP, en el final anterior del sulcus intraparietal, se disparaban cuando el mono veía y manipulaba diferentes tipos de objetos. La mayoría de las neuronas mostraban una preferencia para un tipo particular de objeto sobre los otros probados; por ejemplo, una neurona particular puede dispararse de forma máxima cuando el mono manipula una palanca en una orientación particular, más que cuando manipula una palanca en otra orientación o manipula un tirador. Basándose en las respuestas de manipulación con luz, manipulación con oscuridad, y vista de los objetos sin manipulación, informaban de varias subpoblaciones de neuronas de AIP. Las neuronas «motoras dominantes» respondían sólo cuando ocurría la manipulación (con luz o con oscuridad); las neuronas «visuales dominantes» respondían sólo cuando el objeto era visible (con o sin manipulación); y las neuronas «visomotoras» respondían mejor cuando el objeto era visto y manipulado. Como población, las neuronas de AIP eran generalmente selectivas a la forma, tamaño y orientación pero eran relativamente insensibles a la posición (Taira, Mine, Georgopoulos, Murata y Sakata, 1990).

Las neuronas de manipulación de la mano se encontraban también en el área 5 de Brodmann, un área de asociación somatosensorial en el banco anterior del sulcus intraparietal (a lo largo del sulcus desde AIP) (Mountcastle et al., 1975). Mientras, las neuronas de SI estaban más afectadas durante el contacto de la mano, levantamiento y sujeción, las neuronas de CPP (5 y 7) se disparaban más vigorosamente durante la formación de la forma de la mano previa al contacto. Así pues, parece que el córtex del CPP, particularmente en AIP, está involucrado en el reconocimiento de objetos durante la preparación de acciones de la mano hacia ellos y quizás también en la evaluación de si la forma planeada de la mano se corresponde con el objetivo (basado en un feedback visual y/o en un corolario de descarga desde el córtex premotor, Sakata et al. 1995). La evidencia más convincente para un papel crucial de AIP en la forma de la mano previa a coger el objeto viene de estudios de inactivación de AIP (Gallese, Murata, Kaseda, Niki y Sakata, 1994). Cuando el procesamiento se interrumpe químicamente, el mono pierde la capacidad para formar la mano contralateral para coger un objeto particular, aunque la capacidad para alcanzar la localización de un objeto permanece intacta y, después de contactar el objeto, el mono puede corregir la forma de cogerlo.

1.2.8. Un área para coger objetos en humanos

La sugerencia más fuerte de que los humanos, como los monos, tienen una región cerebral especializada para coger objetos viene de la literatura neuropsicológica. Los pacientes con lesiones en el CPP pueden demostrar ataxia óptica, un déficit en coger objetos y en hacer movimientos para alcanzarlos conocido como síndrome de Balint. Además, estos déficits para coger y alcanzar parecen ser dissociables. En un estudio neuropsicológico, Binkofski y otros (1998) demostraron que los pacientes humanos con lesiones en el sulcus intraparietal anterior mostraban déficits en coger objetos mientras la capacidad de alcanzar permanecía relativamente intacta (igual que la ejecución de los monos con el AIP inactivado).

Los datos de neuroimagen también sugieren una región en humanos especializada en coger objetos. Binkofski (1998) demostró más activación en RMNf para coger objetos comparado con alcanzarlos en sujetos normales en la misma área que, cuando eran lesionados, había conducido a déficits en coger objetos. Sin embargo, otros estudios que intentaban encontrar tal «área para coger objetos» han tenido resultados dispares. Un estudio de TEP informaba de actividad en el CPP anterior con una tarea para coger objetos visualmente guiada (vs. alcanzar objetos) (Faillenot, Toni, Decety, Gregoire y Jeannerod, 1997).

Los resultados más completos son de Culham (2004), que empezó su búsqueda de un área para coger objetos, el área homóloga de AIP, desarrollando un paradigma robusto para estudiar la tarea de coger objetos, examinando la contribución de los componentes somatosensoriales, visuales y de acción. Una vez

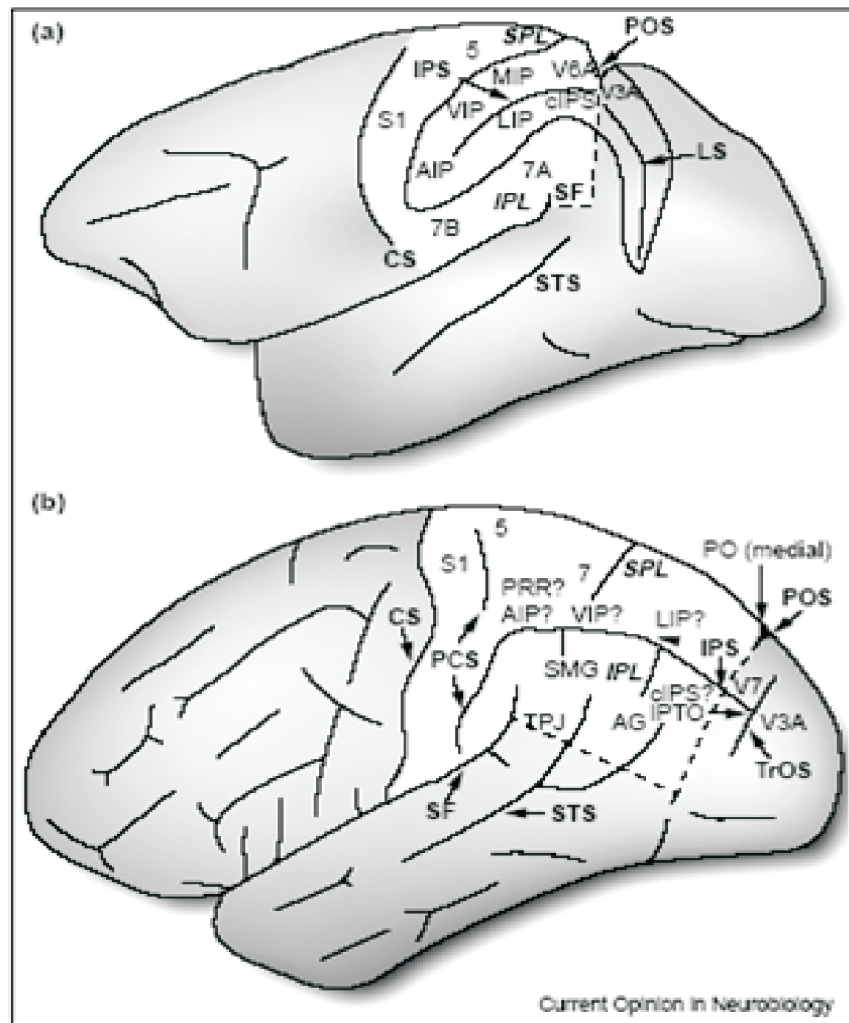


Figura 1.4: Córtex Parietal: (a) cerebro de mono (b) cerebro humano

que el «área para coger objetos» había sido establecida a lo largo de numerosos sujetos individuales, se planteó cuestiones teóricas acerca de su papel en el procesamiento de objetos para la acción, comparado con el área selectiva ventral para los objetos, LO. Otra pregunta que se planteó es el efecto de introducir una demora entre la visión de un objeto y la iniciación del movimiento. Las acciones demoradas eran útiles para disociar la activación debida a la visión y a la acción. Finalmente, investigó si las capacidades que se preservaban para coger objetos de D.F., el paciente con agnosia visual para la forma, pueden ser servidas por el área para coger objetos en humanos. Los resultados confirmaron sus expectativas.

1.2.9. Nuevas investigaciones sobre un área para coger objetos en humanos

Las investigaciones de los últimos años llevan a postular un área, el córtex superior parieto-occipital (SPOC) como el área que codifica el espacio cercano. Hay un cuerpo de evidencias creciente de que SPOC está involucrado en el guiado de los movimientos del brazo (De Jong, Van der Graaf y Paans, 2001; Astafiev, Shulman, Stanley, Snyder, Van Essen y Corbetta, 2003; Cavina-Pratesi, Galletti, Fattori, Quinlan, Goodale y Culham, 2006). SPOC parece codificar no sólo las acciones del brazo, sino las intenciones de movimiento, como indica la respuesta a objetos en tres dimensiones incluso cuando no se realiza la tarea (Cavina-Pratesi et. al, 2006). Las lesiones en ese área están asociadas con déficits severos para alcanzar objetos observados en la ataxia óptica (Karnath y Perenin, 2005). Experimentos de resonancia magnética nuclear funcional realizados por Gallivan

y otros (2009), muestran una mayor activación de SPOC en presencia de objetos que se pueden alcanzar frente a los que no, incluso cuando no se requería ninguna acción, lo que da soporte a postular que los objetos se codifican en el cerebro humano para la acción, independientemente de la percepción.

Capítulo 2

El perceptrón multicapa como máquina de aprendizaje

2.1. Definición de Conexionismo

La aparición de la Ciencia Cognitiva se ha localizado a menudo en un Simposio sobre Teoría de la Información ocurrido en 1956, del 10 al 12 de Septiembre en el Massachussets Institute of Technology (M.I.T.) (Gardner, 1985). En dicho Simposio, varios investigadores de disciplinas distintas dialogaron acerca de la comunicación y de las ciencias humanas. Hubo tres intervenciones en particular, que con el paso de los años, alcanzaron una especial relevancia, la de Miller: «El mágico número 7», la de Chomsky: «Tres modelos de lenguaje» y la de Newell y Simon: «Teoría de las máquinas lógicas». Uno de los resultados

de la reunión fue la creencia de que para comprender la cognición humana hay que combinar los esfuerzos de disciplinas diferentes, una idea que también había expresado Norbert Wiener (1948), el padre de la cibernética.

Ese esfuerzo interdisciplinar común para estudiar la mente humana se puede definir como Ciencia Cognitiva. La idea básica que da sentido a esa disciplina común es que la mente es un procesador de información. Eso incluye información y los procesos que la manipulan. La primera línea de investigación posterior se caracterizó por plantear como hipótesis un sistema físico de símbolos, línea propuesta por Newell y Simon (1976), que podemos llamar la aproximación «clásica» de la ciencia cognitiva. Explicándolo de forma muy básica, la hipótesis establece que la cognición está basada en patrones de información, que estos patrones de información pueden ser representados con símbolos, y que estos símbolos pueden ser manipulados. De ahí, que se ha asumido muy a menudo que la arquitectura de la mente es la arquitectura de ordenadores de estilo Von Neumann, postura asumida por ejemplo por Pylyshyn (1984), debido a que los ordenadores manipulan símbolos. La segunda línea de investigación, el conexionismo, no tuvo en primera instancia la importancia de la otra línea. A menudo ha sido vista como una aproximación radicalmente diferente para estudiar la arquitectura de la mente, remarcando aspectos de la cognición humana pobremente comprendida por las aproximaciones tradicionales (por ejemplo degradación elegante, memoria dirigida por el contenido). Para entender las diferencias, hagamos una breve revisión de las propiedades del conexionismo presentando diferencias con la visión tradicional tradicional.

El conexionismo, dentro de la Ciencia Cognitiva, es una teoría de procesamiento de la información. Si los sistemas clásicos usan reglas explícitas, muy a menudo inspiradas en una lógica determinada, organizadas en una jerarquía para manipular símbolos de una forma serial, los sistemas conexionistas realizan un procesamiento en paralelo de sub-símbolos, usando propiedades estadísticas en lugar de reglas lógicas para transformar la información. Los modelos conexionistas se basan en el conocimiento de la neurofisiología del cerebro e intentan incorporar aquellas propiedades funcionales necesarias para la cognición.

Las propiedades funcionales necesarias del cerebro que son requeridas para el procesamiento de información no son totalmente conocidas. La aproximación conexionista ha usado preferentemente como bloque básico del cerebro la neurona. Dependiendo de los distintos autores y modelos, se han considerado distintos elementos o componentes básicos de una red neuronal. Rumelhart, Hinton y McClelland (1986) elaboraron una lista de ocho propiedades que son esenciales en los modelos que ellos definieron de Procesamiento Distribuido en Paralelo (PDP):

- Un conjunto de unidades de procesamiento.
- Un estado de activación.
- Una función de salida para cada unidad.
- Un patrón de conectividad entre unidades.
- Una regla de propagación para propagar los patrones de actividad a través de la red de conectividades.

- Una regla de activación para combinar las entradas incidiendo sobre una unidad con el estado corriente de esa unidad para producir un nuevo nivel de activación para la unidad.
- Una regla de aprendizaje por la que los patrones de conectividad son modificados por la experiencia.
- Un entorno dentro del cual el sistema debe operar.

Bechtel y Abramsen (1991) redujeron la lista anterior a cuatro propiedades que distinguen los diferentes tipos de arquitectura conexionista. Estas cuatro propiedades son:

1. La conectividad de unidades,
2. La función activación de las unidades,
3. La naturaleza del procedimiento de aprendizaje que modifica la conexión entre las unidades, y
4. La interpretación semántica de la red.

Las propiedades anteriores de los modelos conexionistas pueden ser resumidas en tres principios básicos. Primero, las señales son procesadas por unidades simples. Segundo, las unidades de procesamiento están conectadas de forma paralela con otras unidades de procesamiento. Tercero, las conexiones entre unidades de procesamiento tienen un peso o valor. Estos tres principios son muy

amplios en su descripción para acomodar todos los aspectos del conexionismo. Explicitaremos tales principios.

El procesamiento de señales acompaña la recepción, transformación y transmisión de información, pero esas señales pueden ser llevadas por medios eléctricos, químicos o mecánicos. Además las señales pueden ser suministradas por un estímulo externo (tal como una luz incidiendo en la retina) o de otras unidades de procesamiento. Las unidades de procesamiento pueden referirse a neuronas, funciones matemáticas, u otro tipo de mecanismo. La información puede estar codificada en las unidades localmente o de forma distribuida. Las conexiones entre unidades pueden ser o no masivamente paralelas en el sentido que una unidad puede estar conectada a cualquier otra unidad. Las conexiones pueden ser «feedforward» (i.e. señales que van en una sola dirección) o «multidireccionales» (i.e. señales que pasan en las dos direcciones). Finalmente, los pesos asociados con las conexiones pueden ser «precableados», aprendidos, o ambos. Los pesos representan la fuerza de las conexiones (excitatorias o inhibitorias) entre dos unidades.

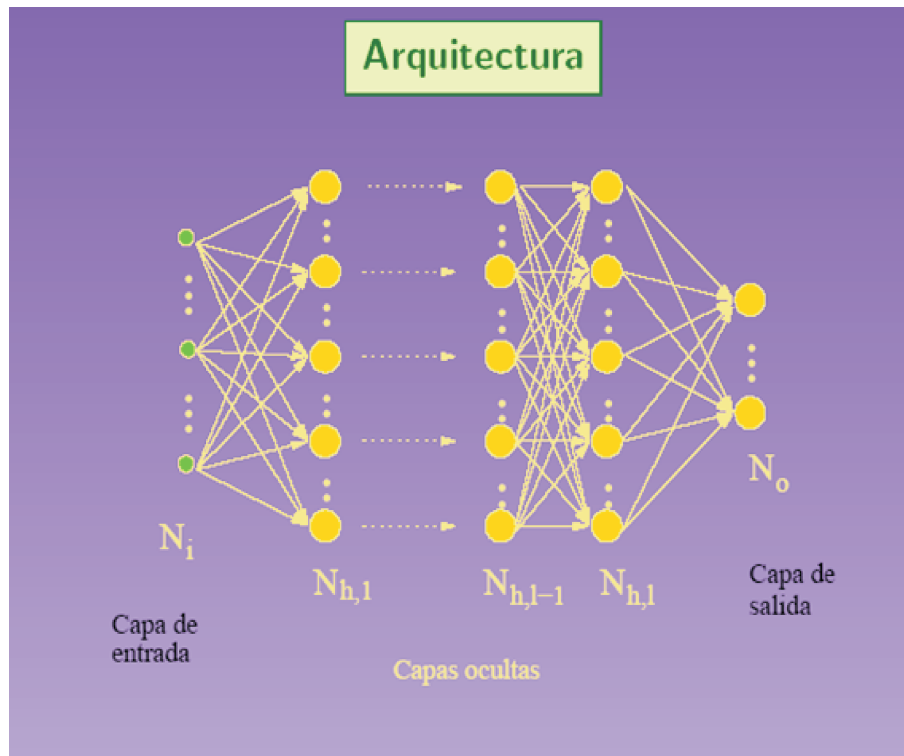


Figura 2.1: Arquitectura genérica de red neuronal MLFN

2.2. La arquitectura conexionista genérica: MLFN

2.2.1. Definición de SLFN y MFLN

A mediados de los años 80, el campo resurge con fuerza, y ello se debe a la gran influencia de los modelos de Procesamiento Distribuido en Paralelo (PDP). Esta arquitectura es básicamente una red multicapa entrenada con el algoritmo de Rumelhart, Hinton y Williams (1986). La red genérica de PDP es probablemente la más conocida y la más ampliamente usada. Muchas de las aplicaciones prácticas

que han surgido del campo de las redes neuronales han usado el algoritmo de retropropagación («backpropagation»). El algoritmo es especialmente adecuado para las tareas de aproximación de funciones y para problemas de clasificación de patrones.

Veamos más extensamente la red estándar PDP. Un perceptrón (un perceptrón con una capa oculta o más) tiene poder en principio para resolver cualquier problema de clasificación de patrones expresado en notación binaria, mientras una red con menos profundidad está limitada en su poder computacional. Este incremento en capacidad computacional deriva del hecho de que una red multicapa puede dividir teóricamente el espacio de patrones en un número arbitrario de regiones de decisión, como demostrara Lipmann (1987). Además, puede demostrarse que tales redes son aproximadores universales de funciones, esto es, que son capaces de resolver problemas de aproximación de funciones hasta un grado arbitrario de precisión como demostrasen Cybenko (1989) y Hornik, Stichcombe y White (1989) para una sola capa oculta. Mhaskar (1993) lo demostró para redes de varias capas ocultas. Estos resultados son específicos de la arquitectura de red, y no del algoritmo usado para entrenar las redes.

Así, se necesita hacer una distinción entre la arquitectura de la red y la regla de aprendizaje. Esto sirve para un propósito doble. Primero, nos permite hacer afirmaciones acerca del poder computacional de las redes independientemente del procedimiento de entrenamiento usado. Segundo, se puede evaluar la regla de aprendizaje independientemente de la arquitectura de la red. La consecuencia de hacer esta distinción entre arquitectura y regla de aprendizaje es que permite:

- Plantear cuestiones acerca de la «plausibilidad biológica» de ciertos algoritmos de aprendizaje (por ejemplo, del método de «backpropagation») sin comprometer la interpretación y los resultados finales de la red entrenada.
- Determinar si las diferencias en la ejecución de una red son debidas a las discrepancias de la arquitectura o a modificaciones del algoritmo de aprendizaje.

Por ello, definiremos primero la arquitectura conexionista genérica, y luego veremos la regla de aprendizaje.

El bloque constructivo para la arquitectura conexionista genérica es la neurona artificial. Las propiedades funcionales de la neurona artificial mimetizan las de las neuronas; esto es, la neurona recibe e integra información, procesa esta información, y transmite la nueva información. Matemáticamente, se puede expresar una neurona k como:

$$u_k = \sum_{i=1}^d w_{ki} x_i$$

donde x_1, \dots, x_d son las señales de entrada, y $w_{k1}, w_{k2}, \dots, w_{kd}$ son los pesos sinápticos de la neurona k .

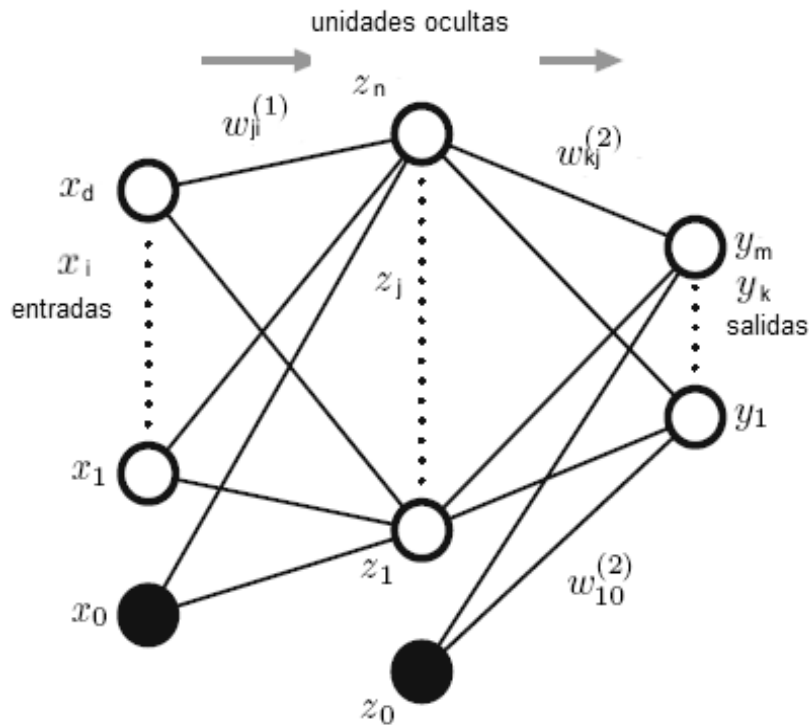


Figura 2.2: Red neuronal SLFN (Una capa oculta)

La función de salida de la neurona es una función no lineal de su entrada y está expresada en la ecuación (2.1). Hay que hacer notar que las reglas de entrenamiento usadas habitualmente requieren que la función de activación de la neurona artificial sea diferenciable y monótona. Consecuentemente, la función más común de activación usada es la función logística o sigmoide, que comprende un rango de entrada de la red de tal forma que la señal de salida esté entre 0 y 1. Esta función permite a la red procesar señales grandes sin saturación y señales pequeñas sin atenuación excesiva. Así, en la ecuación (2.1), y_k es la salida de la neurona, u_k es el resultado de las entradas a dicha neurona, y w_k^0 es el sesgo («bias») de la unidad, el cual funciona como una especie de umbral.

$$y_k = f(u_k) = (1 + e^{-u_k + w_k^0})^{-1} \quad (2.1)$$

El modelo básico de red neuronal, MLP («Multilayer Perceptrón») o MLFN («Multilayer Feedforward Network») puede ser descrito como una serie de transformaciones funcionales. Primero se construyen n combinaciones afines de las variables de entrada x_1, \dots, x_d en la forma:

$$a_j = \sum_{i=1}^d w_{ji}^1 x_i + w_j^0$$

donde $j = 1, \dots, n$. El superíndice indica que los parámetros correspondientes están en la primera capa de la red. Nos referiremos a los parámetros w_{ji}^1 como *pesos* y a los parámetros w_j^0 como *sesgos*. Las cantidades a_j se conocen como *unidades de activación*. Cada una de ellas se transforma usando una *función de activación* $g(\cdot)$ diferenciable y no lineal para dar

$$z_j = g(a_j)$$

Estas cantidades corresponden a las salidas y se conocen como *unidades ocultas*. Las funciones no lineales $g(\cdot)$ son generalmente funciones sigmoideas como la función logística. Estos valores se vuelven a combinar linealmente para dar *unidades de activación de salida*.

$$y_k = \sum_{j=1}^n w_{kj}^2 z_j$$

donde $k = 1, \dots, m$, y m es el número total de salidas. Por último, las unidades de activación de salida se transforman usando una función de activación apropiada para dar un conjunto de salidas de red y_k . La elección de la función de activación se determina por la naturaleza de los datos y las variables de salida.

La forma final de una red neuronal «feedforward» de una sola capa será:

$$y_k = N_k(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^n w_{kj}^2 g\left(\sum_{i=1}^d w_{ji}^1 x_i + w_j^0\right) \quad (2.2)$$

donde el conjunto de todos los pesos y los sesgos se han agrupado juntos en un vector \mathbf{w} . Así, el modelo de red neuronal es simplemente una función no lineal de un conjunto de variables de entrada x_i a un conjunto de variables de salida y_k controladas por un vector \mathbf{w} de parámetros ajustables. Esta función se puede representar en forma de un diagrama de red como se muestra en la figura 2.2. El proceso de evaluar (2.2) se puede interpretar como una *propagación hacia delante* («feedforward») de la información a través de la red. La arquitectura de la figura 2.2 tiene una capa de unidades ocultas. Puede ser fácilmente generalizada, considerando capas adicionales de procesamiento consistentes cada una de ellas de una combinación lineal de pesos seguida de una transformación usando una función de activación no lineal. A la red de la figura 2.2 le llamaremos SLFN, red neuronal «feedforward» de una capa oculta. La definición formal es:

Sea σ una función sigmoïdal acotada, es decir, $\lim_{t \rightarrow -\infty} \sigma(t) = 0$ y $\lim_{t \rightarrow \infty} \sigma(t) = 1$. Una red neuronal «feedforward» de una sola capa (SLFN) es una función dada por

$$y = N(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^n w_j^2 \sigma(\mathbf{w}_j^1 \cdot \mathbf{x} + w_j^0) \quad (2.3)$$

donde $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{w}_j^1 \in \mathbb{R}^d$, y $w_j^2, w_j^0 \in \mathbb{R}$, $j = 1, \dots, n$. A \mathbf{w}_j^1 y w_j^2 se les llama *pesos interiores* y *exteriores* respectivamente. A los parámetros w_j^0 se les llamará *sesgos*.

A partir de (2.3) se pueden definir inductivamente redes neuronales multicapa de forma que no existan bucles cerrados. En este caso hablamos de una arquitectura «hacia delante» y la red resultante se denomina red multicapa «feedforward» (MLFN) (ver, por ejemplo (Mhaskar, 1993)). Se denota por TLFN una MLFN que tiene dos capas de neuronas ocultas, como en la figura 2.3.

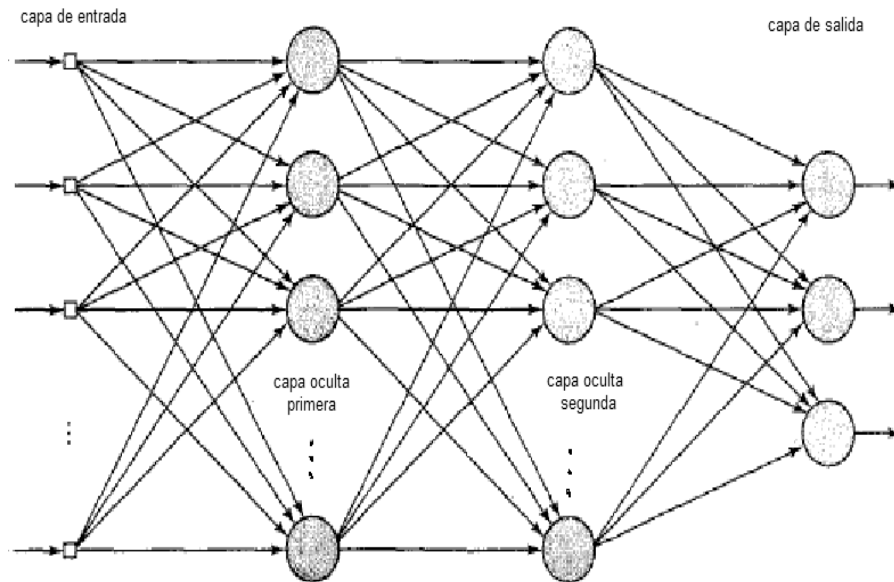


Figura 2.3: Red neuronal TLFN

2.2.2. Propiedades de Aproximación

Esta sección resume algunos resultados fundamentales sobre la capacidad de las redes «feedforward» MLFN para aproximar funciones continuas. El problema de aproximación de f mediante una red neuronal se traduce en la práctica en un problema de interpolación, es decir, se trata de encontrar unos pesos \mathbf{w} tales que $N(\mathbf{x}_j, \mathbf{w}) = t_j$, $j = 1, \dots, M$. El principal resultado de aproximación es que una red «feedforward» con una capa de entrada, una capa intermedia con un número suficiente de unidades ocultas con función de activación sigmoideal y una capa de salida de unidades con función de activación lineal es capaz de aproximar cualquier función continua con la precisión deseada.

Las primeras demostraciones rigurosas para la universalidad de redes neuronales «feedforward» empleando funciones de activación sigmoideas, así como otras funciones de activación más generales, fueron realizadas por Cybenko, Hornik et al. y Funahashi en 1989. El teorema de Cybenko dice: «Sea g cualquier función continua de tipo sigmoideal (por ejemplo $g(\xi) = 1/(1 + e^{-\xi})$). Entonces, dada cualquier función real continua sobre $[0, 1]^d$ (o cualquier otro subconjunto compacto de \mathbb{R}^n) y $\varepsilon > 0$, existen vectores $\mathbf{w}_1^1, \mathbf{w}_2^1, \dots, \mathbf{w}_n^1$ y una función parametrizada $N(\mathbf{x}, \mathbf{w}) : [0, 1]^d \rightarrow \mathbb{R}$ tal que

$$|N(\mathbf{x}, \mathbf{w}) - f(\mathbf{x})| < \varepsilon \quad \forall \mathbf{x} \in [0, 1]^d$$

donde

$$N(\mathbf{x}, \mathbf{w}) = \sum_{j=1}^n w_j^2 g(\mathbf{w}_j^1 \cdot \mathbf{x} + w_j^0)$$

y $\mathbf{w}_j^1 \in \mathbb{R}^d, w_j^2, w_j^0 \in \mathbb{R}, j = 1, \dots, n$.» Hornik et al. (1989) y Funahashi (1989) de forma independiente demostraron teoremas similares estableciendo que una red neuronal con una capa oculta es capaz de aproximar uniformemente cualquier función continua multivariada con cualquier grado de precisión. Hornik et al. (1990) demostraron otro resultado importante respecto a la capacidad de aproximación de redes neuronales «feedforward» con unidades de activación sigmoideas en la capa oculta. Demostraron que estas redes aproximan también las derivadas de dicha función. La universalidad de las redes neuronales de una capa oculta con unidades de activación no sigmoideas como aproximantes fue probada formalmente por Stinchcombe y White (1989). Hornik (1991) probó que se puede tener la propiedad de aproximación universal usando funciones de activación que

sean continuas, acotadas y no constantes. Leshno, Lin y Pinkus (1993) extendieron estos resultados mostrando que redes neuronales con funciones de activación acotadas y continuas por trozos en las unidades ocultas son aproximadores universales si y sólo si la función de activación no es un polinomio. Las propiedades de aproximación de las redes multicapa (MLFN) fueron estudiadas por Mhaskar (1993).

2.3. Panorámica histórica de las MLFN. La Retro-propagación como el primer método iterativo para entrenar MLFNs

Además de la capacidad de aproximación con la que cuentan las MLFNs o MLPs (perceptrones multicapa), una propiedad importante de dichas redes es que pueden ser entrenadas por medio de ejemplos. Varias técnicas de entrenamiento habían sido propuestas anteriormente (e.g. aprendizaje supervisado y genético de Selfridge (1959) ; regla de refuerzo de Rosenblatt (1958); regla delta de Widrow y Hoff (1960)), pero con una sola capa de pesos. Lo que el conexionismo necesitaba era una regla de aprendizaje general para las redes de un número de capas arbitrario. De esta manera, una red relativamente simple con un algoritmo de aprendizaje puede ser aplicada a un amplio rango de tareas.

A continuación describimos la regla delta generalizada, también conocida como el algoritmo de retropropagación («backpropagation») estándar. La regla

delta generalizada puede ser considerada como una de las contribuciones más significativas de la investigación conexionista ya que ha permitido el entrenamiento de las redes multicapa. De hecho, el trabajo de Rumelhart et al. (1986) es a menudo citado como el catalizador para el resurgimiento de la investigación conexionista en la última mitad de los años 80. El procedimiento de entrenamiento es conocido comúnmente como retropropagación del error o «backpropagation». Aunque Rumelhart et al. (1986) han recibido a menudo el crédito de haberla descubierto, la regla de aprendizaje fue derivada previamente en tres ocasiones independientes, primero por Werbos en 1974, luego por Parker en 1982, y finalmente por LeCun en 1986. Werbos (1995) hace notar que dado que la retropropagación se usa en muchas aplicaciones diferentes, su definición actual se ha confundido y adulterado muy a menudo.

La retropropagación es uno de los métodos más simples y generales para entrenar redes neuronales multicapa. Se presenta un conjunto de ejemplos a las unidades de entrada y se pasan las señales a través de la red hasta llegar a las unidades de salida. Se considera que una función objetivo a aproximar $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ está definida por un conjunto de ejemplos: $\{(\mathbf{x}_j, t_j), j = 1, \dots, M\}$. Aquí se supone que $t_j = f(\mathbf{x}_j)$. El aprendizaje supervisado consiste en presentar un ejemplo de entrada, compararlo con los ejemplos *objetivo*, y cambiar los parámetros de red (pesos) en orden a hacer más parecida la salida de la red con el ejemplo *objetivo*. Se comienza con una red no entrenada, se presenta un ejemplo de entrenamiento a las unidades de entrada y se determina la salida de la red. El criterio de error es alguna función escalar de los pesos que se debe minimizar cuando las salidas de la red se acerquen las salidas deseadas. Los pesos se ajustan para reducir la medida del error. Consideramos el *error de entrenamiento* sobre

un ejemplo como la suma sobre las unidades de salida de la diferencia cuadrática entre la salida deseada t_j y la salida real $N(\mathbf{x}_j, \mathbf{w})$:

$$E(\mathbf{w}) = \sum_{j=1}^M (t_j - N(\mathbf{x}_j, \mathbf{w}))^2$$

La regla de aprendizaje de retropropagación se basa en el método de descenso del gradiente. Los pesos se inicializan con valores aleatorios, y se cambian en la dirección que reduce el error:

$$\Delta \mathbf{w} = -\eta \frac{\partial E}{\partial \mathbf{w}}$$

o en componentes:

$$\Delta w_{sr} = -\eta \frac{\partial E}{\partial w_{sr}}$$

donde η es la *tasa de aprendizaje*, e indica el tamaño relativo del cambio en los pesos. El método iterativo es:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta \mathbf{w}(k), \quad k = 1, 2, \dots$$

El valor $\mathbf{w}(0)$ suele tomarse de forma aleatoria. En esencia el algoritmo de retropropagación es el descenso por gradiente utilizando la regla de la cadena

para calcular derivadas. Este algoritmo se puede aplicar a redes «feedforward» en las que:

- Las unidades de entrada se conectan a las de salida directamente (así como a las unidades ocultas).
- Hay más de una capa oculta.
- Hay diferentes funciones de activación para diferentes capas.
- Cada neurona tiene su propia función de activación.
- La tasa de aprendizaje no depende de neuronas individuales.

2.4. Inconvenientes de la Retropropagación

Ahora vamos a enumerar una serie de ventajas y desventajas de la arquitectura de redes multicapa (MLP) entrenadas por «backpropagation»:

1. Existencia de Mínimos locales. La retropropagación de errores de las redes multicapa «feedforward» no garantiza la convergencia a la solución óptima y puede llevar a mínimos locales en la superficie del error.
2. Ejecución rápida. La red opera de atrás a delante («feedforward»), sin ninguna reentrada o feedback. Así, una vez que ha aprendido, la red funciona muy rápida, con un tiempo proporcional al número de capas.

3. Velocidad de aprendizaje. La velocidad de aprendizaje puede ser muy lenta, implicando miles de iteraciones.
4. Número de neuronas en la capa oculta y aproximación. Un problema que se plantea es que no existe una regla general acerca de cuantas neuronas son apropiadas en las capas ocultas, dado que esto depende del tipo de función a aprender. En la práctica, estas redes son a veces entrenadas con distintos números de neuronas, hasta que el número mínimo requerido para realizar la correspondencia ha sido alcanzado.

El método de retropropagación parece difícil de implementar biológicamente. En contraste, las neuronas reales en el cerebro, probablemente utilizan reglas de aprendizaje más sencillas, en las cuales los errores no se retropropagan hacia atrás. El reto futuro es descubrir como redes más plausibles biológicamente que los modelos tipo «backpropagation» pueden resolver problemas difíciles, y examinar las propiedades de estas redes como medio de comprender lo que hace realmente el cerebro.

5. Regla de aprendizaje no local. La regla de aprendizaje no es local para cada neurona. La información requerida para cambiar cada peso sináptico no está disponible en la entrada presináptica y la activación postsináptica. El error para cada neurona de la capa oculta se calcula basándose en los errores de todas las neuronas de la capa de salida con los que la neurona está conectada, teniendo en cuenta los pesos sinápticos más relevantes. No es plausible suponer que la información correcta para suministrar el error adecuado para cada neurona de la capa oculta se propague hacia atrás entre neuronas reales. Una neurona de una capa oculta debería saber «como», o

recibir información acerca de los errores de todas las neuronas con las que está conectada, y de sus pesos sinápticos y de sus activaciones. Por todo lo anterior, haría falta encontrar un mecanismo en el córtex cerebral para suministrar una señal de error apropiada a cada neurona de salida de la red. En el cerebelo existe una arquitectura donde cada célula de Purkinje recibe una entrada que podría llevar la señal de error (Ito, 1984).

Se abre un campo de investigación para implementar algoritmos que realicen aprendizaje local en las neuronas. De tales algoritmos podemos esperar que requieran más neuronas y más capas ocultas que las redes de retropropagación. El reto es encontrar un tipo de red biológicamente plausible que opere con un gran número de neuronas, y que desarrolle una aproximación adecuada.

2.5. Métodos para Construir una Máquina de Aprendizaje Neuronal.

Definimos una máquina de aprendizaje neuronal (NLM) como cualquier función con parámetros ajustables cuyos componentes elementales son MLFNs sin procesos de retroalimentación. Las MLFNs pueden estar combinadas linealmente, multiplicadas o compuestas. Esto permite generalizar el uso de redes MLFN. Las redes MLFN son aproximadores universales para funciones definidas sobre subconjuntos de \mathbb{R}^d , como ya hemos visto.

Dada una función a aproximar $f : \mathbb{R}^d \longrightarrow \mathbb{R}^m$ (función objetivo), se define:

Modelo de Aprendizaje Inicial (ILM) es una función (no necesariamente NLM) de la forma $i : \mathbb{R}^d \longrightarrow \mathbb{R}^m$ con parámetros ajustables.

Máquina de Aprendizaje Neuronal Entrenada (TNLM) es una NLM obtenida a partir de un modelo de aprendizaje inicial que interpola o aproxima en un determinado contexto, la función objetivo f .

Un buen modelo para aprendizaje neuronal en procesos donde los datos de entrada son cambiantes y las necesidades en tiempo real son altas, como son los procesos que se dan en el cerebro para representar objetos en 3 dimensiones debiera cumplir las siguientes condiciones:

- A. Muestreo automático del espacio de entrada para obtener resultados de aprendizaje óptimo. Este proceso debería estar determinado por el problema de aprendizaje considerado y la precisión requerida.
- B. Una construcción automática de la estructura topológica de la red desde los datos de entrenamiento.
- C. Velocidad de aprendizaje rápida.
- D. Buenas propiedades de aproximación (generalización) de las TNLM.

Dividimos los métodos existentes para construir redes neuronales en dos grupos:

1. **Métodos directos de construcción de TNLMs.** En estos métodos, el modelo de aprendizaje inicial es una máquina de aprendizaje neuronal

(NLM) con parámetros ajustables que tras el entrenamiento genera una TNLM para el problema objetivo. La TNLM puede tener una arquitectura (topología de las conexiones y de neuronas) diferente del ILM. Los métodos directos pueden ser clasificados de acuerdo a su algoritmo de entrenamiento en dos grupos:

a) *Métodos Iterativos*. Estos procedimientos pueden ser clasificados de acuerdo al tipo de red TNLM obtenida:

- La TNLM es una MLFN. Los métodos de entrenamiento son métodos de descenso del gradiente (como el de retropropagación de errores «backpropagation») o algoritmos relacionados, métodos de optimización global, etc. En algunos métodos la topología del ILM evoluciona hacia una topología diferente de la red TNLM. Esto se consigue por medio de algoritmos evolutivos (algoritmos genéticos, estrategias evolutivas, etc.) (Yao, 1999).
- La red TNLM es una red neuronal modular (MNN) (Sharkey, 1999). En una red MNN, la tarea se divide en un número de subproblemas por unidades de puertas lógicas («gating»). Cada subproblema se aproxima por una red neuronal independiente (especialista). Las MNNs importantes en aproximación de funciones son:
 - Mezcla de Expertos (Jacobs, Jordan, Nowlan y Hinton, 1991; Jordan y Jacobs, 1994). Estas redes MNN son entrenadas por el algoritmo de Maximización de la Expectativa (EM) u otros métodos iterativos (Szymanski, Lemmon y Bett, 1998).

- Contrapropagación Extendida o Redes «Spline» (Hecht-Nielsen, 1987; Walter y Ritter, 1995). Estas redes aprenden en dos pasos. Primeramente, las unidades de puerta lógica de la capa oculta producen un agrupamiento («clustering») del espacio de entrada que se corresponde con una división de Voronoi d -dimensional. En una segunda fase las unidades de «spline» se entrenan por algoritmos de tipo descenso por gradiente.
- MNNs autogeneradas (Chen, Yang, Yu y Chi, 1997). Usan el principio «divide y vencerás» para generar un árbol neuronal, el cual sitúa redes MLFN en los nodos de tipo hoja. El entrenamiento consiste en una división sucesiva del conjunto de datos. Esta división se realiza usando como criterio de desdoblamiento el error de retropropagación correspondiente a una arquitectura MLFN fija.

b) *Métodos Constructivos*. Estos métodos intentan resolver el problema de interpolación usando interpolantes aproximados (MLFNs), los cuales tienen una expresión analítica predeterminada que no requiere entrenamiento. En el caso $d = 1$, algunos de estos interpolantes aproximados pueden conseguir una aproximación uniforme cuando el número de neuronas es bastante grande (Cardaliaguet y Euvrard, 1992; Llanas y Sáinz, 2006).

Una red SLFN con a lo más n unidades ocultas puede aprender n observaciones distintas con error cero. Huang, Zhu y Siew (2006a) propusieron un algoritmo de entrenamiento rápido, el cual elige al azar

los pesos internos y los sesgos ocultos, y determina analíticamente los pesos exteriores de la red SLFN. Huang probó en 2003, usando un método constructivo, que una red TLFN puede aprender n ejemplos distintos $(\mathbf{x}_i, \mathbf{t}_i)$ donde $\mathbf{x}_i \in \mathbb{R}^d$ y $\mathbf{t}_i \in \mathbb{R}^m$ con error arbitrario, y máximo de $2\sqrt{(m+2)n}$ ($\ll n$) neuronas ocultas. Basándose en este resultado Huang et al. (2006b) propusieron un algoritmo de aprendizaje con capacidad de aprendizaje en tiempo real («Extreme learning machine»).

Los métodos directos basados en entrenamiento iterativo no satisfacen las condiciones A y C. La condición B sólo se verifica por algunos de estos métodos, pero esto implica un tiempo de aprendizaje adicional significativo. El entrenamiento de redes modulares de tipo mezcla de expertos locales es más rápido que el correspondiente a MLFNs entrenadas por métodos iterativos, pero su capacidad de generalización no es significativamente mejor que la obtenida por una MLFN entrenada por retropropagación («backpropagation») (Ronco y Gawthrop, 1995). Por otro lado, la determinación del número de expertos para el aprendizaje de una tarea dada con poca información a priori es una cuestión abierta (Nguyen, Abbass y Mc.Kay, 1993; Redhead y Heywood, 2005). En el caso de redes «spline», la elección de datos de entrenamiento para obtener una buena teselación no es una tarea trivial.

Los métodos directos constructivos tienen una velocidad de aprendizaje rápida y por ello satisfacen la condición C, pero no cumplen A, y D no está garantizada cuando $d > 1$.

2. **Métodos indirectos de construcción de TNLMs.** En estos métodos el ILM no es una NLM. Dada una función objetivo f el método de construcción tiene dos fases:

- **Fase 1.** Mediante un algoritmo se pasa del ILM $i(\mathbf{x})$ a una función $I(\mathbf{x})$ que aproxima (interpola) la función objetivo. Ni i ni I son máquinas de aprendizaje neuronal.
- **Fase 2.** Cada componente elemental de $I(\mathbf{x})$ es aproximado por una NLM.

El resultado final es una máquina de aprendizaje neuronal entrenada (TNLM).

Para la fase 1 se han utilizado por ejemplo, polinomios de Bernstein (Attali y Pagés, 1997; Li, 1996), polinomios de Bernstein-Durrmeyer (Xu y Cao, 2005), aproximación polinomial de mínimos cuadrados (Ritter, 1999), etc. Los métodos indirectos no minimizan el error de interpolación neuronal, por ello el cumplimiento de las condiciones A, B y D debería mejorar. Sin embargo, aunque estos procedimientos han sido usados en demostraciones teóricas (resultados de densidad, velocidad de aproximación, etc.), no han sido aplicados a problemas de la vida real. Esto es debido a las siguientes razones:

- Los métodos constructivos para obtener aproximantes polinómicos tales como los polinomios de Bernstein no convergen suficientemente rápido para funciones continuas y diferenciables.

- Para alcanzar una aproximación uniforme buena es necesario usar polinomios de grado alto. Aunque se demuestra en (Attali y Pagés, 1997) que los polinomios de varias variables pueden ser aproximados uniformemente por una red SLFN con un número finito de neuronas, la construcción de esta red requiere:
 - Resolver sistemas lineales con matrices de tipo Vandermonde, los cuales están severamente mal condicionados cuando el grado de los polinomios es alto (Gaustchi e Inglese, 1988). Una técnica usando polinomios exponenciales para evitar este problema se usó en (Ritter, 1999), pero el procedimiento resultante no controla el tamaño de los pesos y pueden originar inestabilidades numéricas.
 - La computación de derivadas de orden alto de la función de activación. Esta es una tarea difícil, por ejemplo, en el caso de la función logística aparece el problema numérico conocido como «intermediate expression swell» (Minai y Williams, 1993).

Las dificultades anteriores con métodos indirectos podrían ser evitadas si consideramos una aproximación local en lugar de una global y polinomios de grado bajo como aproximante tal y como veremos en el capítulo 5. El algoritmo de entrenamiento de la fase 1, aunque está fuera del contexto de las redes neuronales, puede tener más plausibilidad biológica debido a su sencillez, que muchos algoritmos directos. Por ejemplo, en este trabajo se presenta un algoritmo de desdoblamiento adaptativo que genera automáticamente un árbol de aproximación (topología de la red).

En general, la fase 2 se puede realizar mediante un entrenamiento sencillo que no requiere técnicas de optimización. En este trabajo se prueba constructivamente que cualquier función afín $a : \mathbb{R}^d \longrightarrow \mathbb{R}$ puede ser aproximada por medio de una SLFN con sólo dos neuronas y cuyos pesos están determinados analíticamente.

Capítulo 3

El problema de representar objetos en el espacio para la acción:

Metodología

3.1. Introducción

La revisión de la investigación neurobiológica que hemos realizado en el capítulo 1 nos permite fijar la existencia de una zona en el córtex parietal humano donde se realizan una serie de procesos orientados a la acción sobre los objetos. Esa zona recibe entradas sensoriales a través de la vía dorsal. La representación de los objetos para operar posteriormente sobre ellos tiene características distintas de la representación que tiene que ver con la vía ventral.

Nuestra línea de trabajo es buscar una representación con un formato adecuado para la acción posterior, distinta de otras representaciones más relacionadas con la percepción. Esa representación de los objetos debe poder expresar propiedades métricas en el espacio susceptibles de ser utilizadas posteriormente en el córtex premotor y motor. Después de haber hecho un recorrido histórico por los modelos de redes neuronales y su significado cognitivo, hemos hecho una revisión del modelo de «backpropagation», ya que representa un paradigma fundamental en el mundo de las redes neuronales. A partir de este modelo han surgido múltiples investigaciones en todos los aspectos posibles, tanto computacionales, por ejemplo en la mejora del algoritmo básico, como en su uso en procesos cognitivos. Eso ha hecho que se convierta en una referencia dentro del campo de investigación, pero que posee muchas limitaciones. Una de las críticas clásicas al algoritmo de retropropagación es su implausibilidad biológica, ya que opera de forma global. Por otro lado, aunque ha demostrado eficiencia a la hora de resolver problemas de aprendizaje en muchos ámbitos distintos, presenta puntos débiles desde el punto de vista matemático. Como veremos en los capítulos posteriores, a pesar de que existen modificaciones que han mejorado su eficacia, este algoritmo es demasiado lento para resolver problemas tales como la representación geométrica. El objetivo fundamental de nuestra investigación es construir un modelo de representación de objetos en el espacio a través de propiedades métricas. De lo dicho en el capítulo 1, dicho correlato podría estar situado en el córtex parietal posterior, en la zona colindante con el córtex occipital (SPOC). El modelo debe responder a la exigencia computacional de representar cambios en tiempo real en el espacio. Todas estas exigencias nos llevan a la búsqueda de nuevos modelos de red y nuevos algoritmos de aprendizaje. Nuestro

objetivo es comprender un proceso cognitivo de representación de los objetos, a través de propiedades métricas, mediante un modelo computacional.

3.2. Metodología

El problema de representar las propiedades de objetos en el espacio que sean útiles para coger objetos, un problema resuelto en el córtex parietal humano, se considera en esta tesis como un proceso independiente y que es susceptible de ser explicado independientemente de otros procesos. Las investigaciones neurobiológicas de los últimos años permiten una separación clara entre la acción sobre los objetos y la percepción de estos. En nuestro trabajo se asume que el procesamiento de las propiedades de los objetos necesario para la acción sobre ellos se realiza mediante una computación. Esa computación se realiza físicamente por un conjunto de neuronas. Dado que no podemos implementar exactamente lo que un cerebro realiza, no disponemos de neuronas reales ni de la posibilidad de aislar totalmente un proceso, nuestro método es «simular» dicho proceso. Utilizamos la hipótesis de que esa simulación es lo suficientemente válida para poder explicar lo que ocurre en un cerebro real. Las estrategias de simulación son diversas. Por ejemplo, algunos investigadores utilizan «hardware» específico para reproducir una red neuronal.

En la arquitectura de información que manejamos, los datos entran por la capa de entrada y proceden hasta la salida de la red. Se compone de varias capas, y al menos hay una capa intermedia. Ya hemos hablado de los problemas

que plantea una arquitectura clásica con el algoritmo de retropropagación en el capítulo anterior. Para resolver esas dificultades se abre la vía de permitir que la red cambie de arquitectura en función del problema, construyendo un árbol variable. Los modelos habituales consideran una arquitectura fija, lo cual limita las posibilidades de aprendizaje. Este tema será discutido en el capítulo 5. La construcción de redes neuronales que sean capaces de simular adecuadamente el proceso que queremos modelizar plantea diversos problemas que se tratarán de resolver en esta tesis. Es necesario probar y comparar distintas «arquitecturas» de red, así como distintos «algoritmos» que sean capaces de resolver adecuadamente el problema computacional planteado. En nuestro caso la «arquitectura» creada debe ser capaz de representar objetos con distinto grado de complejidad en tiempo real, que es lo que hace el cerebro. Otro problema planteado que debemos resolver es la gran dificultad computacional que presenta el espacio en tres dimensiones, y la necesidad de que las redes neuronales puedan adaptarse a la complejidad de los objetos representados. Todos estos problemas que surgen al intentar modelizar el proceso de representación de objetos en el espacio se abordan en la siguiente sección.

3.3. Planteamiento del problema

Los distintos problemas que se abordan en este trabajo necesitan una explicación cuidadosa: El proceso de representación de los objetos para la acción es realizado por el cerebro mediante conjuntos de neuronas. Lo que hace el cerebro es computar mediante esas neuronas unas determinadas propiedades de

los objetos que son necesarias para esa representación, y además lo tiene que hacer en tiempo real, es decir, la computación debe ser rápida, sobre una gran cantidad de datos sensoriales, utilizando conjuntos amplios de neuronas. Los datos de entrada los suministra el sistema sensorial, fundamentalmente la visión, aunque no olvidemos que en el córtex parietal se integran distintas fuentes sensoriales. Las redes neuronales de nuestro modelo utilizan esos datos para construir en tiempo real una representación en 3 dimensiones de los campos de distancia de un objeto, necesarios para coger dicho objeto. La red neuronal tiene dos fases, en la fase I aprende, en la II ejecuta. En la primera fase (aprendizaje) se necesitan integrar rápidamente los datos de entrada, de tal forma que en la fase II la ejecución de la red neuronal sea precisa y rápida. El cerebro resuelve este problema de forma automática y no consciente, es un procesamiento en tiempo real.

3.4. Requisitos de los algoritmos

Nuestra hipótesis de trabajo es que los modelos «feedforward» actuando en distintas capas, pueden realizar el procesamiento anterior, pero utilizando algoritmos distintos de aprendizaje al de «backpropagation». Hemos visto las limitaciones del algoritmo de retropropagación en el capítulo 2. Necesitamos nuevos algoritmos que cumplan:

1. Localidad. La computación debe ser local. Este es un requisito de plausibilidad biológica.

2. Rapidez de aprendizaje. El entrenamiento de MLFN con algoritmos clásicos es muy lento. Hablamos de procesos de representación de objetos que están implementados biológicamente y que son absolutamente necesarios para la supervivencia de la especie.
3. Capacidad de Aproximación. El modelo de red, una vez suministrado un conjunto de ejemplos, debe ser capaz de generalizar esas propiedades a todo el objeto.
4. Rapidez de ejecución. Cada objeto, una vez que se han representado las propiedades necesarias para la acción, debe ser procesado rápidamente, en tiempo real.
5. Precisión. La representación debe ser lo suficientemente buena para permitir posteriormente coger el objeto.
6. La topología de la red debe tener una organización biológicamente plausible. Es decir, debe tener analogía con la anatomía del cerebro.

El proceso cognitivo de representar con precisión un objeto en el espacio para actuar sobre él, como la operación de cogerlo adecuadamente, implica representar propiedades geométricas de ese objeto que sean imprescindibles para la acción. Muchas de las investigaciones que se han hecho sobre la vía dorsal llegan a la conclusión de que en esta vía se representan propiedades que no están centradas en el objeto, sino en distintas zonas del sujeto, ya sean ojos, cuerpo o mano (Cohen y Andersen, 2002). La localización de objetos en el espacio en relación al cuerpo humano se ha formalizado con distintos modelos matemáticos. Esto se ha venido haciendo en el campo de la robótica desde hace tiempo, y hay un gran número

de estudios donde se utilizan algoritmos para calcular la distancia entre objetos y partes del cuerpo humano, fundamentalmente la mano. Sin embargo, ese tipo de aproximaciones no se han construido desde el punto de vista de la plausibilidad cognitiva. La ventaja de las redes neuronales desde el punto de vista cognitivo es su analogía con el funcionamiento cerebral, lo que las convierte en candidatas para la modelización de un proceso cognitivo. En nuestro caso, queremos modelizar el proceso cognitivo de representación de los campos de distancia de un objeto en el espacio, que permita localizar su posición y que permita posteriormente coger dicho objeto. El modelo incluye propiedades métricas que no están centradas en el objeto, sino que son relaciones geométricas entre la mano y el objeto. A través de esa relación se construyen las propiedades necesarias para la acción de coger. La información de entrada es la situación de la mano y del objeto. La red neuronal va a aprender esa relación respecto a todo el objeto, a través de ejemplos y de un proceso de generalización, es decir, tras la fase de aprendizaje la red actúa ante cualquier nueva situación. Todo esto se puede resumir en varias hipótesis clave:

1. La representación de los objetos en el espacio, a través de su relación con el cuerpo, como la distancia a la mano, es necesaria para coger un objeto.
2. El cerebro realiza una computación que construye con sus propios medios ese procedimiento necesario para implementar el proceso cognitivo. Esas redes de neuronas cerebrales son las que formaliza la teoría de redes neuronales artificiales.
3. Uno de los modelos, procedente de la teoría de redes neuronales artificiales, la arquitectura «feedforward», tiene capacidad para aproximar cualquier función, debido a que se ha demostrado que las redes neuronales

«feedforward» son aproximadores universales de funciones. Ese es un punto de conexión básico entre las redes neuronales y el proceso cognitivo de localizar y representar las propiedades métricas de objetos en el espacio.

4. Es conocido teóricamente que las redes neuronales artificiales «feedforward» son aproximadores universales de funciones, pero hace falta encontrar algoritmos constructivos y topologías adecuadas que aproximen una función con un grado deseado de precisión y puedan funcionar posteriormente en tiempo real. Esto se puede conseguir mediante árboles neuronales «feedforward».

3.5. Formulación matemática y computacional

El procedimiento de establecer relaciones entre algoritmos y procesos biológicos en diferentes niveles refleja la distinción clásica de David Marr (1982) de múltiples niveles de descripción, originalmente desarrollada para el análisis de los sistemas de visión. El nivel computacional se define por la formulación teórica abstracta del problema computacional a ser resuelto por un sistema, un ejemplo es la identificación o clasificación de un objeto, o el reconocimiento de los gestos de una mano. Aunque Marr lo llame computacional lo identificamos mejor como un nivel matemático abstracto. El nivel algorítmico de Marr especifica los métodos computacionales para la solución del problema independientemente de la arquitectura o hardware específico. El uso de redes neuronales y su plausibilidad biológica responde al nivel de implementación ya que nuestro propósito es asimilarlo al procesamiento cerebral. En nuestro caso, la formalización matemática y la construcción de los algoritmos concretos

estarían dentro del nivel algorítmico de Marr. Cada uno de estos tres niveles en la terminología de David Marr implica un trabajo distinto. Los modelos de redes neuronales, como hemos visto en el capítulo 2 han ido creciendo como formas de analogía e implementación de funciones cognitivas y de formas de trabajo del propio cerebro. En nuestro caso, la utilización de un modelo de red neuronal está en relación con el nivel de implementación, ya que se supone que esa implementación se asimila a la cerebral. Independientemente de la terminología de David Marr, que puede servir como inspiración dentro del campo de la Ciencia Cognitiva, nuestra investigación se va a mover en varios niveles. Nuestra intención es que sean complementarios:

1. Un nivel matemático, demostrando la existencia de modelos neuronales que sean capaces de resolver un problema abstracto de aprendizaje dado un conjunto de ejemplos. Esto es importante, porque no se pueden buscar modelos si antes no es seguro que dada una arquitectura y unos ejemplos, existe un aprendizaje seguro.
2. Un nivel algorítmico. Dado un modelo matemático, existe un algoritmo que resuelve el modelo.
3. Un nivel computacional. El algoritmo se implementa en un lenguaje de programación y se realizan distintos experimentos computacionales, variando las condiciones de aprendizaje y comparando los resultados con otros resultados obtenidos con modelos distintos que a su vez se han implementado computacionalmente. Los resultados se exponen tanto en tablas, como de forma gráfica.

4. Un nivel biológico. El modelo es plausible biológicamente, en el sentido de que no existen restricciones biológicas que lo invaliden.
5. Un nivel cognitivo. El modelo responde a las necesidades de la representación de objetos para un procesamiento orientado a la acción en el córtex parietal posterior.

3.6. Entorno experimental

La realización de los experimentos computacionales y de los programas que implementan los distintos algoritmos que componen esta tesis implica el uso de un entorno informático, entendiendo éste como la suma de un equipamiento «hardware» y «software». En nuestro caso, la implementación que se realiza es una emulación mediante «software» en un ordenador serie convencional de características medias en el mercado. Esto no limita los resultados experimentales, ya que con los mismos algoritmos, el resultado sería mejor con ordenadores más potentes. Una ventaja de usar un equipo informático de características estándar y unas herramientas de «software» comunes es facilitar la comparación de resultados con otros experimentos en el campo. Las herramientas utilizadas para realizar los programas son conocidas dentro del campo de la simulación, el lenguaje de programación C (Kernighan y Ritchie, 1988), que es idóneo para el cálculo numérico. Los gráficos en 2 y 3 dimensiones, así como las gráficas de resultados, se han realizado con el programa Matlab de la empresa Mathworks. Hacemos un pequeño resumen de los elementos de hardware y software:

■ Ordenador con:

- Unidad Central de Proceso: Pentium IV de 2 Gigahertzios (INTEL).
- Memoria RAM de 1 Gigabyte.
- Disco Duro de 80 Gigabytes.
- Sistema Operativo Windows XP (Microsoft).

■ Software:

- Visual C (Microsoft).
- MATLAB (Mathworks).

Capítulo 4

Un enfoque constructivo de la aproximación de funciones mediante redes neuronales

4.1. Introducción

En el capítulo 2 hemos realizado un recorrido por la génesis y el desarrollo de los modelos de redes neuronales artificiales «feedforward». Los modelos clásicos como el de retropropagación del error presentan una serie de limitaciones que es necesario solucionar. Un paso necesario es una adecuada formalización del problema del aprendizaje mediante redes neuronales. Para ello vamos a fijarnos en dos conceptos formales que se plantean con el uso de redes neuronales: la

interpolación y la aproximación de funciones. La interpolación es un método para obtener aproximantes. La mayoría de los métodos de entrenamiento de redes obtienen interpolantes, que en algunas ocasiones son buenos aproximantes, es decir, generalizan bien.

Este capítulo está organizado en seis secciones, incluyendo la introducción. La sección 4.2 introduce el problema de la interpolación mediante redes neuronales y da una demostración nueva y cuantitativa del hecho que $n + 1$ neuronas en la capa oculta de una SLFN pueden aprender $n + 1$ ejemplos distintos con error cero. La sección 4.3 introduce las redes de interpolación aproximada. Estas redes no requieren entrenamiento y pueden interpolar un conjunto arbitrario de ejemplos distintos. Se proporciona un límite superior riguroso del error de interpolación y se muestra que básicamente depende de un parámetro real (A) y del número de neuronas de la red.

La sección 4.4 da una demostración constructiva de la convergencia uniforme por medio de interpolantes aproximados a cualquier función continua de una variable (si ϕ es la función logística sigmoideal). El Lema 8 y el Teorema 3 describen tal red neuronal cuyos pesos son conocidos explícitamente.

Aunque Cardaliaguet y Euvrard (1992) han propuesto también aproximantes con características similares, nuestro enfoque presenta algunas ventajas:

- Proporciona una red neuronal genuina, esto es, la función de activación es sigmoideal. En (Cardaliaguet et. al, 1992) las funciones de activación son «bell-shaped» (en forma de campana) e incluso con la transformación que

proponen en la sección 3.1.4 el resultado es una red neuronal con términos adicionales que dependen del número de neuronas.

- Obtener el número de neuronas n como una función de la precisión ε es una tarea fácil en el Lema 8, pero es mucho más difícil en (Cardaliaguet et. al, 1992).

La sección 4.5 da una demostración nueva del teorema de Cybenko (1989). Aunque se han propuesto otras demostraciones constructivas del teorema de Cybenko, con funciones de activación más generales (Chen, Chen y Liu, 1991; Deba0, 1993; Mhaskar y Michelli, 1992), damos un operador de aproximación más general obtenido a partir de las ai-nets. De esta forma, mostramos una conexión entre interpolación y aproximación en el contexto de las redes neuronales.

Finalmente, en la sección 4.6 se da una demostración constructiva del hecho de que una función afín puede ser aproximada uniformemente por una SLFN que sólo tiene dos neuronas en la capa oculta.

4.2. Interpolación y redes neuronales. El problema de la interpolación exacta con una SLFN

4.2.1. Interpolación y redes neuronales

Un problema de interpolación está determinado por un conjunto de pares ordenados

$$(\mathbf{x}_0, f_0), (\mathbf{x}_1, f_1), \dots, (\mathbf{x}_n, f_n), \quad (4.1)$$

donde $S = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$ es un conjunto de vectores distintos y $\{f_i/i = 0, \dots, n\}$ es un conjunto de números reales.

Decimos que la función $f : \mathbb{R}^d \rightarrow \mathbb{R}$ es un *interpolante exacto* de (4.1) si

$$f(\mathbf{x}_i) = f_i \quad (i = 0, 1, \dots, n).$$

Consideremos el siguiente conjunto de funciones

$$\mathcal{N}_{n+1, \phi}^d = \{N(\mathbf{x}) = \sum_{j=0}^n c_j \phi(\mathbf{w}_j \cdot \mathbf{x} + b_j) / \mathbf{w}_j \in \mathbb{R}^d, c_j, b_j \in \mathbb{R}\},$$

donde $\mathbf{w} \cdot \mathbf{x}$ denota el producto escalar usual de \mathbb{R}^d y ϕ es una función de \mathbb{R} a \mathbb{R} .

Llamamos:

- Redes «feedforward» de una sola capa a los elementos de $\mathcal{N}_{n+1,\phi}^d$.
- Funcion de activación a ϕ .
- Pesos internos a los coeficientes \mathbf{w}_j y b_j .
- Pesos externos a los coeficientes c_j .

Cuando ϕ es una función sigmoidal, esto es, se verifica $\lim_{t \rightarrow -\infty} \phi(t) = 0$ y $\lim_{t \rightarrow \infty} \phi(t) = 1$, llamamos:

- Redes neuronales «feedforward» de una capa a los elementos de $\mathcal{N}_{n+1,\phi}^d$.
- Neurona a cada sumando de $N(\mathbf{x})$.

Las redes neuronales «feedforward» de una sola capa con a lo más $n + 1$ sumandos (elementos de $\mathcal{N}_{n+1,\phi}^d$) pueden aprender $n + 1$ ejemplos distintos (\mathbf{x}_i, f_i) con error cero (interpolantes exactos) y los pesos internos pueden ser elegidos «casi» arbitrariamente.

Se han propuesto varias demostraciones de este hecho:

- Demostraciones de tipo analítico se dan en (Ito, 1996; Ito y Saito, 1996; Ito, 2001; Pinkus, 1999; y Sontag, 1992). Ito y Saito (1996) prueban que si la función de activación es sigmoidal, continúa y no decreciente, entonces la interpolación se puede hacer con pesos internos $\mathbf{w}_j \in S^{d-1}$ ($S^{d-1} \equiv \{\mathbf{x} \in \mathbb{R}^d / x_1^2 + x_2^2 + \dots + x_d^2 = 1\}$).

Pinkus (1999) prueba el mismo resultado pero ϕ sólo necesita ser continua en \mathbb{R} y no un polinomio.

- Demostraciones algebraicas, que tienen un carácter constructivo se pueden encontrar en (Huang y Babri, 1998), (Sartori y Antsaklis, 1991), (Shrivastava y Dasgupta, 1990) y (Tamura y Tateishi, 1997). Shrivastava y Dasgupta (1990) dan una demostración para funciones de activación logísticas sigmoidales y entonces, prueban el teorema (por medio de un procedimiento asintótico) para funciones de activación ϕ , tales que $\phi(0) \neq 0$ y ϕ^{-1} exista. Sartori y Antsaklis (1991) consideran que ϕ es cualquier función no lineal. Huang y Babri (1998) critican la demostración anterior porque consideran que tiene una validez menos general. Dan una demostración para funciones de activación arbitrarias no lineales acotadas, las cuales tienen un límite en el infinito, esto es, existe $\lim_{t \rightarrow \infty} \phi(t)$ o $\lim_{t \rightarrow -\infty} \phi(t)$.

Con respecto al problema de encontrar los pesos de la red, todas las demostraciones algebraicas fijan los pesos internos de una forma más o menos arbitraria y los pesos externos se obtienen resolviendo un sistema lineal $(n+1) \times (n+1)$.

Otras aproximaciones directas para obtener los pesos son más complejas, por ejemplo, Barhen, Cogswell y Protopescu (2000), conectan una capa virtual de entrada a la capa de entrada nominal mediante una función especial de transferencia no lineal y a la primera capa por sinápsis regulares (lineales). Una secuencia de descomposiciones de valor singular de direcciones alternadas se usa para determinar con precisión los pesos entre capas.

Li (2002), considera que ϕ es m veces continuamente diferenciable sobre un intervalo conteniendo 0 y $\phi^{(k)}(0) \neq 0$ ($0 \leq k \leq m-1$). Da un algoritmo para interpolar aproximadamente n ejemplos que consiste en los siguientes pasos:

- Resolver un sistema lineal $(n+1) \times (n+1)$.
- Encontrar la inversa de una matriz de Vandermonde $(n+1) \times (n+1)$.
- Estimar el máximo de una función sobre un intervalo.

Los procedimientos citados anteriormente pueden ser poco efectivos, especialmente, cuando el número de neuronas es grande. Incluso los métodos (algebraicos) más simples requieren invertir una matriz densa $(n+1) \times (n+1)$. Por ello, nos planteamos el siguiente problema:

Problema I: *¿Hay una forma de encontrar los pesos de una red neuronal interpolante sin entrenamiento?*

Sea ε un número positivo real, decimos que la función $g : \mathbb{R}^d \rightarrow \mathbb{R}$ es un *interpolante ε -aproximado* de (4.1) si

$$|g(\mathbf{x}_i) - f_i| < \varepsilon \quad (i = 0, 1, \dots, n).$$

Dada una función arbitraria g , definimos el *error de interpolación* de g relativo al problema 4.1 como:

$$\max_{i=0,1,\dots,n} |g(\mathbf{x}_i) - f_i|.$$

Sontag (1992) utiliza interpolantes aproximados como una herramienta para estudiar la interpolación exacta. Probó que si existen interpolantes aproximados de precisión arbitraria en un espacio lineal de funciones, entonces se puede obtener un interpolante exacto en ese espacio (Lemma 9.1). Además, Sontag da una demostración del siguiente hecho:

«Si ϕ es sigmoideal, continua y existe un punto c tal que $\phi'(c) \neq 0$, entonces un problema de interpolación con $2n + 1$ ejemplos se puede aproximar con precisión arbitraria por una red con $n + 1$ neuronas».

Aunque esta demostración (basada en el principio de inducción) se puede hacer constructiva, la determinación de los pesos requiere resolver n veces un conjunto de ecuaciones de una variable, en el caso más favorable.

Por otro lado, la mayor parte de los métodos de aproximación por medio de redes neuronales se reducen a la búsqueda de interpolantes aproximados. Estos se obtienen normalmente usando:

- Técnicas de optimización local (descenso por gradiente, métodos de entrenamiento de segundo orden, etc.).
- Procedimientos de optimización global (optimización estocástica, métodos basados en homotopía, etc.).

La conexión entre interpolantes aproximados y aproximación uniforme en el contexto de la Teoría del Aprendizaje Probabilístico (computacional) se puede encontrar en (Anthony, 1994), y (Anthony y Bartlett, 1999).

Podemos enunciar el problema siguiente:

Problema II: *Existencia y construcción de interpolantes neuronales ε -aproximados para valores arbitrarios de ε .*

Es claro que una solución a este problema da un procedimiento para resolver el Problema I en la práctica.

En este capítulo damos una solución al Problema II, esto es, para cualquier problema de interpolación (4.1) y una precisión arbitraria ε , mostramos un método constructivo para obtener una familia de interpolantes ε -aproximados en $\mathcal{N}_{n+1,\phi}^d$ (si ϕ es sigmoideal y no decreciente).

4.2.2. El problema de la interpolación exacta con una SLFN

Sea $\mathcal{P} = \{x_0 = a, x_1, \dots, x_n = b\}$ cualquier partición del intervalo finito $[a, b] \subset \mathbb{R}$. Queremos encontrar una red neuronal N que interpole exactamente los datos

$$(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n),$$

donde $f_i \in \mathbb{R}$ ($i = 0, 1, \dots, n$), esto es

$$N(x_i) = f_i \quad (i = 0, 1, \dots, n).$$

Se definen las siguientes cantidades

$$\Delta_0 \equiv 0,$$

$$\Delta_i \equiv x_i - x_{i-1} \quad (i = 1, \dots, n),$$

$$M \equiv \max_{i=1, \dots, n} \Delta_i,$$

$$m \equiv \min_{i=1, \dots, n} \Delta_i,$$

$$r \equiv m/M,$$

$$\Delta_{ij} \equiv \Delta_i + \Delta_{i+1} + \dots + \Delta_j \quad (\text{if } i < j),$$

$$\Delta_{ii} \equiv \Delta_i \quad (i = 0, 1, \dots, n).$$

De las definiciones anteriores, tenemos

$$0 < r \leq 1,$$

$$\frac{\Delta_i}{\Delta_j} \geq r, \quad (i, j = 1, \dots, n). \quad (4.2)$$

A partir de ahora, consideraremos que $\sigma(t)$ es una función sigmoïdal no decreciente, esto es:

1. $\lim_{t \rightarrow \infty} \sigma(t) = 1.$
2. $\lim_{t \rightarrow -\infty} \sigma(t) = 0.$
3. If $t_1 \leq t_2$, then $\sigma(t_1) \leq \sigma(t_2).$

La tercera condición parece escasamente restrictiva en la práctica, de hecho en (Duch y Jankowski, 1999) se dan siete ejemplos de funciones sigmoïdales y todas ellas son no decrecientes.

Buscamos una función de tipo:

$$N(x) = \sum_{j=0}^n c_j \sigma(w_j x + b_j), \quad (4.3)$$

tal que

$$\sum_{j=0}^n c_j \sigma(w_j x_i + b_j) = f_i \quad (i = 0, 1, \dots, n). \quad (4.4)$$

O, en forma vectorial

$$M\mathbf{c} = \mathbf{f}$$

Los pesos internos de la red se determinan por el siguiente procedimiento:

Sea A una variable positiva real. Sean α y β constantes positivas y negativas, respectivamente. Para $j = 0, \dots, n-1$ consideramos los siguientes sistemas

$$\begin{aligned}w_j x_j + b_j &= \alpha A, \\w_j x_{j+1} + b_j &= \beta A,\end{aligned}$$

y

$$\begin{aligned}w_n x_{n-1} + b_n &= (2\alpha - \beta)A, \\w_n x_n + b_n &= \alpha A.\end{aligned}$$

Esto es, para cada neurona j (excepto la última) podemos elegir su valor en x_j cerca de c_j y su valor en x_{j+1} cerca de 0 cuando A es suficientemente grande. La solución de los sistemas anteriores es como sigue

$$\begin{aligned}
w_j &= \frac{\beta - \alpha}{x_{j+1} - x_j} A \quad (j = 0, 1, \dots, n-1), \\
b_j &= \frac{\alpha x_{j+1} - \beta x_j}{x_{j+1} - x_j} A \quad (j = 0, 1, \dots, n-1), \\
w_n &= \frac{\beta - \alpha}{x_n - x_{n-1}} A, \\
b_n &= \frac{(2\alpha - \beta)x_n - \alpha x_{n-1}}{x_n - x_{n-1}} A.
\end{aligned}$$

Una red neuronal de tipo (4.3), que satisface (4.4) con pesos internos dados por las expresiones anteriores, se llamará *ei-net* (red de interpolación exacta) para los puntos $((x_0, f_0), (x_1, f_1), \dots, (x_n, f_n))$. De ahora en adelante, la denotaremos por $N_e(x, A)$.

Si reemplazamos cada x_j por su valor concreto tenemos

$$\begin{aligned}
w_j &= \frac{\beta - \alpha}{\Delta_{j+1}} A \quad (j = 0, 1, \dots, n-1), \\
b_j &= \left(\alpha + \frac{(\alpha - \beta)(a + \Delta_0 + \Delta_1 + \dots + \Delta_j)}{\Delta_{j+1}} \right) A \quad (j = 0, 1, \dots, n-1), \\
w_n &= \frac{\beta - \alpha}{\Delta_n} A, \\
b_n &= \left(\alpha + \frac{(\alpha - \beta)b}{\Delta_n} \right) A.
\end{aligned}$$

De aquí se tiene que las entradas de la matriz M son:

$$\begin{aligned}
M_{in} &= \sigma \left(\left(\alpha + (\alpha - \beta) \frac{\Delta_{i+1n}}{\Delta_n} \right) A \right) & (i = 0, 1, \dots, n-1), \\
M_{ij} &= \sigma \left(\left(\alpha + (\alpha - \beta) \frac{\Delta_{i+1j}}{\Delta_{j+1}} \right) A \right) & (i < j < n), \\
M_{ii} &= \sigma(\alpha A) & (i = 0, 1, \dots, n), \\
M_{i+1i} &= \sigma(\beta A) & (i = 0, 1, \dots, n-1), \\
M_{ij} &= \sigma \left(\left(\beta + (\beta - \alpha) \frac{\Delta_{j+2i}}{\Delta_{j+1}} \right) A \right) & (j \leq i-2, i = 2, \dots, n).
\end{aligned}$$

Introduzcamos ahora algunas definiciones previas y lemas.

Si S es una matriz cuadrada denotaremos por $\det(S)$ su determinante y por S^t su traspuesta.

Lema 1 Sea $S = (\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_n)$ una matriz cuadrada con vectores columna \mathbf{s}_j . Si definimos

$$S^d \equiv (\mathbf{s}_0 - \mathbf{s}_1, \mathbf{s}_1 - \mathbf{s}_2, \dots, \mathbf{s}_{n-1} - \mathbf{s}_n, \mathbf{s}_n),$$

tenemos que

$$\det(S) = \det(S^d)$$

Demostración. Aplicando las propiedades de los determinantes.

□

Una matriz cuadrada $S = (S_{ij})$ se denomina estrictamente diagonal dominante si

$$|S_{ii}| > \sum_{\substack{j=0 \\ j \neq i}}^n |S_{ij}|$$

para $i = 0, 1, \dots, n$.

Lema 2 ((Lascaux y Theodor, 1986)[p.70]) *Las matrices diagonalmente dominantes son invertibles.*

Lema 3 Sean a_0, a_1, \dots, a_n , $n+1$ números reales arbitrarios. Definimos $a_{n+1} \equiv 0$ y la sucesión finita

$$a_0 - a_1, a_1 - a_2, \dots, a_{n-1} - a_n, a_n - a_{n+1}.$$

Si $k \in \{0, 1, \dots, n\}$ tenemos

$$a_k - a_{k+1} > \sum_{\substack{i=0 \\ i \neq k}}^n (a_i - a_{i+1}),$$

si y sólo si

$$a_k > a_{k+1} + \frac{a_0}{2}. \quad (4.5)$$

Demostración. Si $k = 0$

$$\sum_{i=1}^n (a_i - a_{i+1}) = a_1.$$

Si $k = n$

$$\sum_{i=0}^{n-1} (a_i - a_{i+1}) = a_0 - a_n.$$

Si $n \neq 0$ y $k \neq 0$

$$\sum_{\substack{i=0 \\ i \neq k}}^n (a_i - a_{i+1}) = a_0 - a_k + a_{k+1}.$$

El segundo término en las ecuaciones anteriores es menor que $a_k - a_{k+1}$ si y sólo si se verifica (4.5).

□

Podemos ahora enunciar el

Teorema 1 *Existe un número real A^* tal que si $A > A^*$, entonces la matriz M es invertible.*

Demostración. De ahora en adelante, denotaremos $\alpha - \beta$ por ω .

A partir de los Lemas 1 y 2 es suficiente estudiar cuando $(M^t)^d$ es estrictamente diagonal dominante.

Como las filas de la matriz M^t son sucesiones finitas estrictamente decrecientes podemos aplicar el Lema 3 para determinar las condiciones sobre A, α y β que hagan $(M^t)^d$ estrictamente diagonal dominante.

Se deben dar las siguientes $n + 1$ condiciones

$$\begin{aligned}
\sigma(\alpha A) &> 2\sigma(\beta A), \\
2\sigma(\alpha A) &> 2\sigma(\beta A) + \sigma\left(\left(\alpha + \omega \frac{\Delta_1}{\Delta_2}\right) A\right), \\
2\sigma(\alpha A) &> 2\sigma(\beta A) + \sigma\left(\left(\alpha + \omega \frac{\Delta_{12}}{\Delta_3}\right) A\right), \\
&\dots \dots \dots, \\
2\sigma(\alpha A) &> 2\sigma(\beta A) + \sigma\left(\left(\alpha + \omega \frac{\Delta_{1n-1}}{\Delta_n}\right) A\right), \\
2\sigma(\alpha A) &> \sigma\left(\left(\alpha + \omega \frac{\Delta_{1n}}{\Delta_n}\right) A\right).
\end{aligned}$$

Definamos

$$\begin{aligned}
g(A) &\equiv 2\sigma(\alpha A), \\
g_0(A) &\equiv 4\sigma(\beta A), \\
g_1(A) &= 2\sigma(\beta A) + \sigma\left(\left(\alpha + \omega \frac{\Delta_1}{\Delta_2}\right)A\right), \\
&\dots \dots \dots, \\
g_{n-1}(A) &\equiv 2\sigma(\beta A) + \sigma\left(\left(\alpha + \omega \frac{\Delta_{1n-1}}{\Delta_n}\right)A\right), \\
g_n(A) &\equiv \sigma\left(\left(\alpha + \omega \frac{\Delta_{1n}}{\Delta_n}\right)A\right).
\end{aligned}$$

Se tiene que $\lim_{A \rightarrow \infty} g(A) = 2$, $\lim_{A \rightarrow \infty} g_0(A) = 0$ y $\lim_{A \rightarrow \infty} g_k(A) = 1$ ($k = 1, \dots, n$).

De la definición de límite con

- $\varepsilon = \frac{1}{2}$, existe A' tal que si $A > A'$, entonces $g(A) > \frac{3}{2}$.
- $\varepsilon = \frac{3}{2}$, existe A_0 tal que si $A > A_0$, entonces $g_0(A) < \frac{3}{2}$.
- $\varepsilon = \frac{1}{2}$, existe A_k tal que si $A > A_k$, entonces $g_k(A) < \frac{3}{2}$ para $k = 1, \dots, n-1$.

Por lo tanto, si tomamos $A^* = \max(A', A_0, A_1, \dots, A_{n-1})$, se sigue que si $A > A^*$, entonces

$$g(A) > \frac{3}{2} > g_k(A) \quad (k = 0, 1, \dots, n).$$

□

En el caso de la función de activación logística sigmoidal

$$s(x) = \frac{1}{1 + e^{-x}},$$

una condición suficiente para la existencia de ei-nets es

Corolario 1 *Si la función de activación es la logística sigmoidal, y se da la siguiente desigualdad*

$$e^{\alpha A} \left(\frac{1 - e^{\beta A}}{1 + 3e^{\beta A}} \right) > 1.$$

entonces M es invertible.

Demostración. Las $n + 1$ condiciones del Teorema 1 se cumplen si se satisfacen las siguientes desigualdades

$$\sigma(\alpha A) > 2\sigma(\beta A) \tag{4.6}$$

$$2\sigma(\alpha A) > 2\sigma(\beta A) + \sigma((\alpha + \omega\nu)A) \text{ (para todo } \nu > 0) \tag{4.7}$$

$$2\sigma(\alpha A) > \sigma((\alpha + \omega\tau)A) \text{ (para todo } \tau \geq 1) \tag{4.8}$$

Si $\sigma(x) = s(x)$, (4.6) puede ser escrito como

$$\frac{1}{1 + e^{-\alpha A}} > \frac{2}{1 + e^{-\beta A}},$$

lo cual es equivalente a

$$e^{\omega A} - e^{\alpha A} - 2 > 0. \quad (4.9)$$

La desigualdad (4.7) se puede escribir como

$$\frac{2}{1 + e^{-\alpha A}} > \frac{2}{1 + e^{-\beta A}} + \frac{1}{1 + e^{-(\alpha + \omega v)A}},$$

esto es equivalente a

$$e^{\omega A} - e^{\alpha A} - e^{-\beta A} + \frac{2}{e^{\omega v A}}(e^{-\beta A} - e^{-\alpha A}) - 3 > 0. \quad (4.10)$$

Si se cumplen las desigualdades:

$$e^{\omega A} - e^{\alpha A} - e^{-\beta A} - 3 > 0, \quad (4.11)$$

y

$$e^{-\beta A} - e^{-\alpha A} > 0, \quad (4.12)$$

entonces (4.10) se satisface.

La desigualdad (4.8) se puede escribir como

$$\frac{2}{1+e^{-\alpha A}} > \frac{1}{1+e^{-(\alpha+\omega\tau)A}},$$

Como $A > 0$, esta desigualdad siempre se cumple, en efecto

$$2 + 2e^{-\alpha A}e^{-\omega\tau A} > 2 > 1 + e^{-\alpha A}.$$

Ahora, si

$$e^{\alpha A} \left(\frac{1 - e^{\beta A}}{1 + 3e^{\beta A}} \right) > 1,$$

tenemos

$$e^{\alpha A}(e^{-\beta A} - 1) > e^{-\beta A} + 3,$$

por lo tanto

$$e^{\alpha A}(e^{-\beta A} - 1) - 2 > 1 + e^{-\beta A}.$$

Esto implica (4.9) y (4.11).

Finalmente, como $A > 0$

$$e^{-\beta A} - e^{-\alpha A} > e^{-\beta A} - 1 > 0,$$

y (4.12) se sigue.

□

Notas

1. En el caso común en que $\alpha = 1$ y $\beta = -1$, la desigualdad del Corolario 1 se puede escribir como

$$e^A \left(\frac{1 + e^{-A}}{1 + 3e^{-A}} \right) > 1.$$

Si hacemos el cambio de variable $y = e^A$, esto es equivalente a

$$y^2 - 2y - 3 > 0.$$

Por ello, si $A > \ln(3)$ entonces, M es invertible.

2. El procedimiento para determinar los pesos internos es una modificación del método propuesto en (Huang y Babri, 1998). Podríamos obtener una formulación

más general considerando los sistemas

$$\begin{aligned}w_j x_j + b_j &= g_1(A), \\w_j x_{j+1} + b_j &= g_2(A),\end{aligned}$$

para $j = 0, \dots, n-1$, y

$$\begin{aligned}w_n x_{n-1} + b_n &= 2g_1(A) - g_2(A), \\w_n x_n + b_n &= g_1(A),\end{aligned}$$

donde g_1 y g_2 verifican $\lim_{A \rightarrow +\infty} g_1(A) = +\infty$ y $\lim_{A \rightarrow +\infty} g_2(A) = -\infty$.

Como las condiciones anteriores garantizan que $g_1(A) > g_2(A)$ en un intervalo $(A_p, +\infty)$, para algún $A_p \geq 0$, la demostración del Teorema 1 sigue siendo válida con pequeños cambios.

4.3. Interpolantes aproximados neuronales unidimensionales

Sea $\mathcal{P} = \{x_0 = a, x_1, \dots, x_n = b\}$ cualquier partición del intervalo $[a, b] \subset \mathbb{R}$. Consideremos el problema de interpolación

$$(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n).$$

Definimos la siguiente red neuronal

$$\begin{aligned} N_a(x, A) \equiv & \sum_{j=0}^{n-1} (f_j - f_{j+1}) \sigma \left(\frac{(\beta - \alpha)Ax}{x_{j+1} - x_j} + \frac{\alpha x_{j+1} - \beta x_j}{x_{j+1} - x_j} A \right) + \\ & + f_n \sigma \left(\frac{(\beta - \alpha)Ax}{x_n - x_{n-1}} + \frac{(2\alpha - \beta)x_n - \alpha x_{n-1}}{x_n - x_{n-1}} A \right), \end{aligned} \quad (4.13)$$

donde α y β son constantes reales positivas y negativas respectivamente.

Esta red neuronal se llamará una *ai-net* para los puntos $((x_0, f_0), (x_1, f_1), \dots, (x_n, f_n))$. Probaremos en esta sección que las ai-nets son interpolantes aproximados que están arbitrariamente cerca de la correspondiente ei-net (igual A , α y β) cuando el parámetro A se incrementa. Hacemos notar que $N_a(x, A)$ y $N_e(x, A)$ difieren sólo en los pesos externos, los cuales se dan explícitamente por (4.13) en el caso de una ai-net.

Introducimos ahora algunas definiciones previas y lemas.

En \mathbb{C}^{n+1} introducimos la norma

$$\|\mathbf{z}\|_1 = |z_0| + |z_1| + \dots + |z_n|,$$

donde $|z| = \sqrt{z \bar{z}}$ (\bar{z} es el conjugado del número complejo z).

En el espacio de las matrices complejas $(n+1) \times (n+1)$ la norma de la matriz inducida es

$$\|B\|_1 \equiv \max_{j=0,\dots,n} \left(\sum_{i=0}^n |b_{ij}| \right).$$

Lema 4 (Kress, 1998)[p.80] Sean B y $B + \Delta B$ matrices invertibles complejas. Sean \mathbf{z} y $\mathbf{z} + \Delta \mathbf{z}$ vectores tales que

$$\begin{aligned} B\mathbf{z} &= \mathbf{b}, \\ (B + \Delta B)(\mathbf{z} + \Delta \mathbf{z}) &= \mathbf{b}, \end{aligned}$$

entonces

$$\frac{\|\Delta \mathbf{z}\|_1}{\|\mathbf{z}\|_1} \leq \text{cond}_1(B) \frac{\|\Delta B\|_1}{\|B\|_1} \left(\frac{1}{1 - \text{cond}_1(B) \frac{\|\Delta B\|_1}{\|B\|_1}} \right),$$

donde $\text{cond}_1(B) = \|B\|_1 \|B^{-1}\|_1$ es el número de condicionamiento de B .

Definimos las funciones

$$\delta_1(A) \equiv 1 - \sigma(\alpha A).$$

$$\delta_2(A) \equiv \sigma(\beta A).$$

$$\delta(A) \equiv \max(\delta_1(A), \delta_2(A)).$$

Se cumple que $\lim_{A \rightarrow \infty} \delta(A) = 0$

Teorema 2 *Existe un número real A_1 tal que si $A > A_1$, entonces*

$$|N_e(x, A) - N_a(x, A)| < \frac{2(n+1)\delta(A)}{1-2(n+1)\delta(A)} \left(\sum_{j=0}^{n-1} |f_j - f_{j+1}| + |f_n| \right), \quad (4.14)$$

para todo $x \in [a, b]$.

Demostración. Consideraremos que $A > A^*$ (Teorema 1).

Las redes pueden ser escritas de forma resumida como

$$N_a(x, A) = \sum_{j=0}^n c'_j \sigma(w_j x + b_j),$$

y

$$N_e(x, A) = \sum_{j=0}^n c_j \sigma(w_j x + b_j).$$

Los coeficientes c'_j y c_j son, respectivamente, soluciones de los sistemas

$$U\mathbf{c}' = \mathbf{f} \text{ y } M\mathbf{c} = \mathbf{f},$$

donde

$$U = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & 0 & 1 & \dots & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & 1 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix},$$

y

$$\mathbf{c}' = \begin{pmatrix} c'_0 \\ c'_1 \\ \dots \\ c'_n \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c_0 \\ c_1 \\ \dots \\ c_n \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} f_0 \\ f_1 \\ \dots \\ f_n \end{pmatrix}.$$

Considerando en el Lema 4

$$B = U, \quad \mathbf{z} = \mathbf{c}', \quad M = B + \Delta B, \quad \mathbf{z} + \Delta \mathbf{z} = \mathbf{c}, \quad \mathbf{b} = \mathbf{f},$$

tenemos

$$\frac{\|\mathbf{c} - \mathbf{c}'\|_1}{\|\mathbf{c}'\|_1} \leq \|U^{-1}\|_1 \|M - U\|_1 \left(\frac{1}{1 - \|U^{-1}\|_1 \|M - U\|_1} \right). \quad (4.15)$$

Necesitamos una cota superior de la expresión

$$\|U^{-1}\|_1 \|M - U\|_1.$$

A partir de los valores explícitos de M_{ij} (sección 2), se tiene que si $A > 0$ e $i = 0, 1, \dots, n-1$

$$|M_{in} - 1| = 1 - \sigma \left(\left(\alpha + \frac{\omega \Delta_{i+1n}}{\Delta_n} \right) A \right) < 1 - \sigma(\alpha A) \leq \delta(A),$$

si $i < j < n$

$$|M_{ij} - 1| = 1 - \sigma \left(\left(\alpha + \frac{\omega \Delta_{i+1j}}{\Delta_{j+1}} \right) A \right) < 1 - \sigma(\alpha A) \leq \delta(A),$$

si $i = 0, 1, \dots, n$

$$|M_{ii} - 1| = 1 - \sigma(\alpha A) \leq \delta(A),$$

si $i = 0, 1, \dots, n-1$

$$|M_{i+1i}| = \sigma(\beta A) \leq \delta(A).$$

Finalmente, si $j \leq i-2$ e $i = 2, \dots, n$

$$|M_{ij}| = \sigma \left(\left(\beta - \frac{\omega \Delta_{j+2i}}{\Delta_{j+1}} \right) A \right) < \sigma(\beta A) \leq \delta(A).$$

De las desigualdades anteriores se tiene:

$$\|M - U\|_1 < (n+1)\delta(A).$$

Por otro lado

$$U^{-1} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & -1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & -1 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix},$$

por tanto $\|U^{-1}\|_1 = 2$,

de forma que:

$$\|U^{-1}\|_1 \|M - U\|_1 < 2(n+1)\delta(A). \quad (4.16)$$

Existe un número A' tal que si $A > A'$, entonces $2(n+1)\delta(A) < 1$. Entonces, de (4.15), (4.16) y considerando que la función $g(x) = x/(1-x)$ es estrictamente creciente sobre $(-\infty, 1)$, se tiene que:

$$\|\mathbf{c} - \mathbf{c}'\|_1 < \frac{2(n+1)\delta(A)}{1 - 2(n+1)\delta(A)} \|\mathbf{c}'\|_1. \quad (4.17)$$

Por otro lado

$$\begin{aligned} |N_e(x, A) - N_a(x, A)| &= \left| \sum_{j=0}^n (c_j - c'_j) \sigma(w_j x + b_j) \right| \\ &\leq \sum_{j=0}^n |c_j - c'_j| = \|\mathbf{c} - \mathbf{c}'\|_1. \end{aligned}$$

Finalmente, a partir de (4.17) tenemos que si $A > A_1 = \max(A', A^*)$, entonces

$$|N_e(x, A) - N_a(x, A)| < \frac{2(n+1)\delta(A)}{1 - 2(n+1)\delta(A)} \left(\sum_{j=0}^{n-1} |f_j - f_{j+1}| + |f_n| \right).$$

□

Si los valores f_i son las imágenes de x_i bajo una función real f , el segundo factor en (4.14) se puede acotar como sigue:

- Si f es de variación acotada en $[a, b]$

$$V_a^b(f) + |f(b)|.$$

- Si f es continuamente diferenciable en $[a, b]$

$$\left(\max_{x \in [a, b]} |f'(x)| \right) (b - a) + |f(b)|.$$

- Si f es Lipschitziana en $[a, b]$ y L es una constante Lipschitziana (sección 4.4)

$$L(b - a) + |f(b)|.$$

En el caso de una función de activación logística sigmoideal podemos encontrar una cota más ajustada. Si definimos $\gamma \equiv \min(|\alpha|, |\beta|)$ tenemos

Corolario 2 Si $\sigma(x) = s(x)$, existe un número real A_2 tal que si $A > A_2$, entonces

$$|N_e(x, A) - N_a(x, A)| < \frac{4e^{-\gamma A}}{1 - e^{-\omega r A} - 4e^{-\gamma A}} \left(\sum_{j=0}^{n-1} |f_j - f_{j+1}| + |f_n| \right) \quad (4.18)$$

para todo $x \in [a, b]$.

Demostración. Sea $A > A^* > 0$. Por (4.2), $\frac{\Delta_i}{\Delta_j} \geq r$ para $i, j = 1, \dots, n$, y por ello, la suma de los términos de la columna k de la matriz cuyas entradas son el valor absoluto de las entradas de $M - U$, \mathcal{S}_k ($1 \leq k < n$), se puede acotar como sigue

$$\begin{aligned} 1 - s \left(\left(\alpha + \frac{\omega(\Delta_1 + \dots + \Delta_k)}{\Delta_{k+1}} \right) A \right) &< e^{-(\alpha + k\omega r)A}, \\ 1 - s \left(\left(\alpha + \frac{\omega(\Delta_2 + \dots + \Delta_k)}{\Delta_{k+1}} \right) A \right) &< e^{-(\alpha + (k-1)\omega r)A}, \\ &\dots \dots \dots, \\ 1 - s \left(\left(\alpha + \frac{\omega\Delta_k}{\Delta_{k+1}} \right) A \right) &< e^{-(\alpha + \omega r)A}, \\ 1 - s(\alpha A) &< e^{-\alpha A}, \\ s(\beta A) &< e^{\beta A}, \\ s \left(\left(\beta - \frac{\omega\Delta_{k+2}}{\Delta_{k+1}} \right) A \right) &< e^{(\beta - \omega r)A}, \\ s \left(\left(\beta - \frac{\omega(\Delta_{k+2} + \Delta_{k+3})}{\Delta_{k+1}} \right) A \right) &< e^{(\beta - 2\omega r)A}, \\ &\dots \dots \dots, \\ s \left(\left(\beta - \frac{\omega(\Delta_{k+2} + \Delta_{k+3} + \dots + \Delta_n)}{\Delta_{k+1}} \right) A \right) &< e^{(\beta - (n-k-1)\omega r)A}. \end{aligned}$$

Por tanto

$$\mathcal{S}_k < \frac{e^{-\alpha A}(1 - e^{-(k+1)\omega r A})}{1 - e^{-\omega r A}} + \frac{e^{\beta A}(1 - e^{-(n-k)\omega r A})}{1 - e^{-\omega r A}} < \frac{e^{-\alpha A} + e^{\beta A}}{1 - e^{-\omega r A}}.$$

Como $e^{-\alpha A} \leq e^{-\gamma A}$, $e^{\beta A} \leq e^{-\gamma A}$,

Se tiene:

$$\mathcal{S}_k < \frac{2e^{-\gamma A}}{1 - e^{-\omega r A}}.$$

Esta cota es válida también para las columnas 0 y n .

Por tanto

$$\|M - U\|_1 < \frac{2e^{-\gamma A}}{1 - e^{-\omega r A}}.$$

y

$$\|U^{-1}\|_1 \|M - U\|_1 < \frac{4e^{-\gamma A}}{1 - e^{-\omega r A}}.$$

Existe un número real A'' tal que si $A > A''$, entonces

$$\frac{4e^{-\gamma A}}{1 - e^{-\omega r A}} < 1.$$

Por tanto, si $A > A''$

$$\|\mathbf{c} - \mathbf{c}'\|_1 < \frac{4e^{-\gamma A}}{1 - e^{-\omega r A} - 4e^{-\gamma A}} \|\mathbf{c}'\|_1,$$

y si $A > A_2 = \max(A'', A^*)$, entonces

$$|N_e(x, A) - N_a(x, A)| < \frac{4e^{-\gamma A}}{1 - e^{-\omega r A} - 4e^{-\gamma A}} \left(\sum_{j=0}^{n-1} |f_j - f_{j+1}| + |f_n| \right).$$

□

La Figura 4.1 muestra el gráfico de $N_e(x, A)$ (línea continua) y $N_a(x, A)$ (línea discontinua) cuando A aumenta. Hemos considerado el problema de interpolación (círculos) de la función $\sin(2\pi x)$ en 11 puntos, la función de activación sigmoideal logística, $\alpha = 1$ y $\beta = -1$.

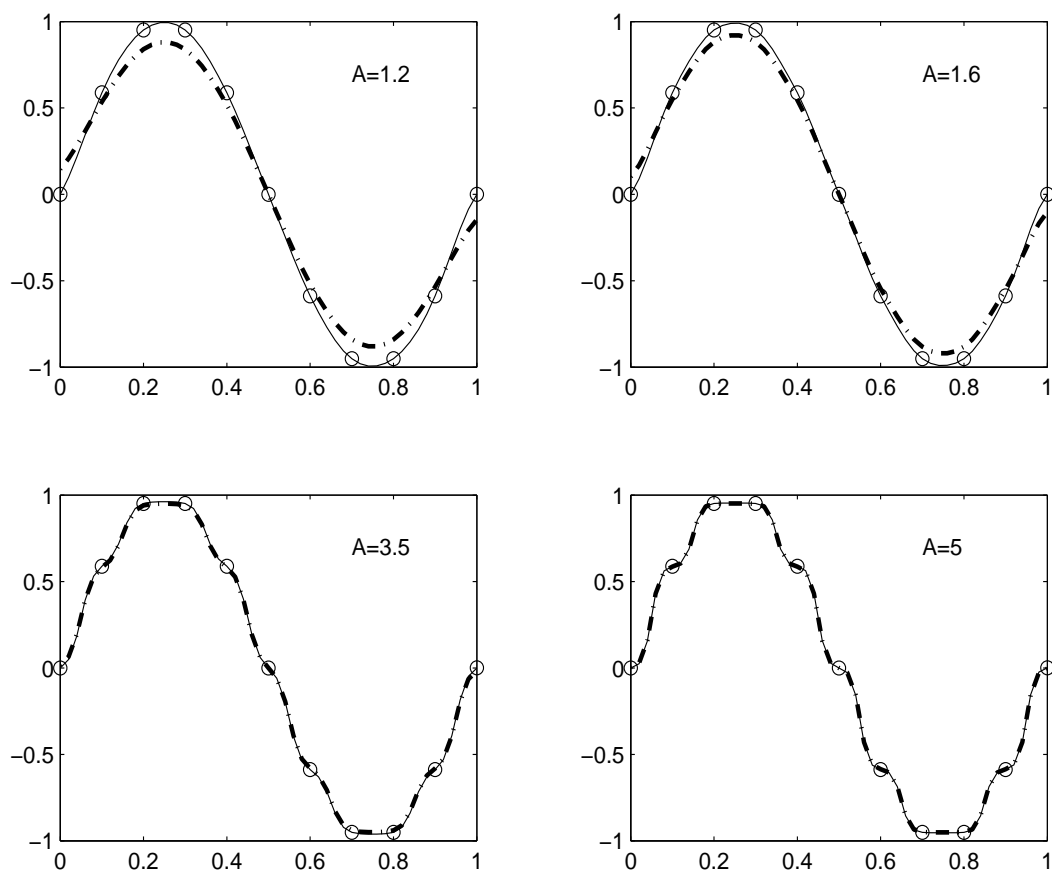


Figura 4.1: N_a en el caso de $\sin(2\pi x)$.

4.4. Aproximación uniforme por medio de interpolantes aproximados

En esta sección usamos los resultados de la sección previa a fin de obtener métodos constructivos para la aproximación uniforme de funciones continuas. Para simplificar la cuestión, nos hemos restringido a la función de activación logística. Introducimos ahora algunas definiciones y resultados auxiliares.

Sea f una función definida en $[a, b]$. El *módulo de continuidad* de f en $[a, b]$, se define para $\delta > 0$ como

$$\omega(f, \delta) = \sup_{\substack{x, y \in [a, b] \\ |x - y| \leq \delta}} |f(x) - f(y)|.$$

Lema 5 ((Rivlin, 1981)[p.14]) f es continua en $[a, b]$ si y sólo si

$$\lim_{\delta \rightarrow 0} \omega(f, \delta) = 0.$$

Lema 6 Sea f una función continua en $[a, b]$. Para cada $\varepsilon > 0$ existe una función lineal por trozos l_ε tal que

$$|f(x) - l_\varepsilon(x)| < \varepsilon \text{ for all } x \in [a, b].$$

Demostración. Por el Lema 5, para cada $\varepsilon > 0$ existe $\eta > 0$ tal que $\omega(f, \delta) < \varepsilon$ si $\delta < \eta$. Consideramos una partición $\mathcal{P} = \{x_0, \dots, x_n\}$ de $[a, b]$ tal que $|x_{i+1} - x_i| < \eta$ para $i = 0, \dots, n-1$ y definimos

$$l_\varepsilon((1-\theta)x_i + \theta x_{i+1}) = (1-\theta)f(x_i) + \theta f(x_{i+1}) \text{ para } \theta \in [0, 1].$$

Entonces, si $x = (1-\theta)x_i + \theta x_{i+1}$, se tiene:

$$|f(x) - l_\varepsilon(x)| \leq (1-\theta)|f(x) - f(x_i)| + \theta|f(x) - f(x_{i+1})| < \varepsilon.$$

□

En esta sección consideramos ai-nets, con las siguientes características:

- La partición de $[a, b]$ es uniforme, es decir, $x_j = a + \frac{b-a}{n}j$ ($j = 0, 1, \dots, n$).
- A depende de n , esto es, $A = A(n)$.
- Los números reales f_j son las imágenes de x_j bajo una función f

$$f_j = f(x_j) = f\left(a + \frac{b-a}{n}j\right).$$

Entonces, si $\alpha > 0$ y $\beta < 0$ son constantes reales podemos definir

$$\begin{aligned} N_a(x, A) &\equiv \sum_{j=0}^{n-1} (f_j - f_{j+1})s \left(\frac{(\beta - \alpha)A(n)x}{x_{j+1} - x_j} + \frac{\alpha x_{j+1} - \beta x_j}{x_{j+1} - x_j} A(n) \right) \\ &+ f_n s \left(\frac{(\beta - \alpha)A(n)x}{x_n - x_{n-1}} + \frac{(2\alpha - \beta)x_n - \alpha x_{n-1}}{x_n - x_{n-1}} A(n) \right). \end{aligned}$$

Lema 7 Sea $s(x)$ la función logística. Se tiene que

$$\text{Si } v > 0, \text{ entonces } \lim_{x \rightarrow \infty} xs(vx)(1 - s(vx)) = 0. \quad (4.19)$$

$$\text{para todo } x \in \mathbb{R} \quad s(x)(1 - s(x)) \leq \frac{1}{4}. \quad (4.20)$$

$$\lim_{x \rightarrow \infty} \frac{\ln(x)}{x} = 0. \quad (4.21)$$

Demostración. Todos estos resultados se pueden obtener fácilmente usando técnicas de Cálculo Diferencial.

□

Una función f es Lipschitziana en $[a, b]$ si existe una constante $L > 0$ tal que

$$|f(x) - f(y)| \leq L|x - y|$$

para todo $x, y \in [a, b]$. Decimos que L es una constante de Lipschitz para f .

Lema 8 Sea f una función Lipschitziana en $[a, b]$. Para todo $\varepsilon > 0$, existe una función $A(n)$ y un número natural N tal que

$$|f(x) - N_a(x, A(n))| < \varepsilon$$

para todo $n > N$ y para todo $x \in [a, b]$.

Demostración. Sea $x \in [a, b]$, y supongamos que $x \in [x_J, x_{J+1}]$, entonces

$$\begin{aligned}
|f(x) - N_a(x, A)| &= |f(x) - f(x_J) + f(x_J) - N_a(x_J, A) + N_a(x_J, A) - N_a(x, A)| \\
&\leq |f(x) - f(x_J)| + |f(x_J) - N_a(x_J, A)| \\
&\quad + |N_a(x_J, A) - N_a(x, A)| \\
&= |f(x) - f(x_J)| + |N_e(x_J, A) - N_a(x_J, A)| \\
&\quad + |N_a(x_J, A) - N_a(x, A)| \\
&\leq \frac{L(b-a)}{n} + |N_e(x_J, A) - N_a(x_J, A)| + |N_a(x_J, A) - N_a(x, A)|. \tag{4.22}
\end{aligned}$$

De (4.18), si $A > A_2$

$$|N_e(x, A) - N_a(x, A)| < \frac{4e^{-\gamma A}}{1 - e^{-\omega A} - 4e^{-\gamma A}} (L(b-a) + |f(b)|). \tag{4.23}$$

El tercer sumando en (4.22), se puede acotar si computamos la derivada de $N_a(x, A)$ para $x \in (x_J, x_{J+1})$.

Si usamos la notación

$$s^d(w_j x + b_j) \equiv s(w_j x + b_j)(1 - s(w_j x + b_j)),$$

y $0 < J < n - 1$, tenemos

$$\begin{aligned}
|N'_a(x, A)| &= \left| \sum_{j=0}^{J-1} \frac{\omega A(f_j - f_{j+1})}{x_j - x_{j+1}} s^d(w_j x + b_j) + \frac{\omega A(f_J - f_{J+1})}{x_J - x_{J+1}} s^d(w_J x + b_J) \right. \\
&\quad \left. + \sum_{j=J+1}^{n-1} \frac{\omega A(f_j - f_{j+1})}{x_j - x_{j+1}} s^d(w_j x + b_j) - \frac{f(b)\omega A n}{b-a} s^d(w_n x + b_n) \right| \\
&\leq \sum_{j=0}^{J-1} \left| \frac{f_j - f_{j+1}}{x_j - x_{j+1}} \right| \omega A s^d(w_j x + b_j) + \left| \frac{f_J - f_{J+1}}{x_J - x_{J+1}} \right| \omega A s^d(w_J x + b_J) \\
&\quad + \sum_{j=J+1}^{n-1} \left| \frac{f_j - f_{j+1}}{x_j - x_{j+1}} \right| \omega A s^d(w_j x + b_j) + \frac{|f(b)|\omega A n}{b-a} s^d(w_n x + b_n) \\
&\leq \omega n A L s(\gamma A)(1 - s(\gamma A)) + \frac{\omega A L}{4} + \frac{|f(b)|\omega A n}{b-a} s(\gamma A)(1 - s(\gamma A)).
\end{aligned}$$

Esta desigualdad es también válida para $J = 0$ y $J = n - 1$ y está basada en los siguientes hechos:

- Si $x \notin [\beta A, \alpha A]$ entonces $s(x)(1 - s(x)) \leq s(\gamma A)(1 - s(\gamma A))$.
- (4.20).
- La Lipschitzianidad de f .

Ahora podemos dar una acotación del tercer sumando en (4.22). Se tiene para algún $x^* \in (x_J, x_{J+1})$

$$\begin{aligned}
|N_a(x, A) - N_a(x_J, A)| &= |N'_a(x^*, A)| |x - x_J| \\
&\leq (\omega n A L s(\gamma A)(1 - s(\gamma A)) + \frac{\omega A L}{4} \\
&\quad + \frac{|f(b)| \omega A n}{b - a} s(\gamma A)(1 - s(\gamma A))) \frac{b - a}{n} \\
&\leq \omega(L(b - a) + |f(b)|) A s(\gamma A)(1 - s(\gamma A)) \\
&\quad + \frac{\omega A L(b - a)}{4n}.
\end{aligned}$$

Se debe encontrar un número N tal que si $n > N$, (4.23) y las siguientes desigualdades

$$\frac{4e^{-\gamma A}}{1 - e^{-\omega A} - 4e^{-\gamma A}}(L(b - a) + |f(b)|) < \frac{\varepsilon}{4}, \quad (4.24)$$

$$\frac{\omega A L(b - a)}{4n} < \frac{\varepsilon}{4}, \quad (4.25)$$

$$\omega(L(b - a) + |f(b)|) A s(\gamma A)(1 - s(\gamma A)) < \frac{\varepsilon}{4}, \quad (4.26)$$

$$\frac{L(b - a)}{n} < \frac{\varepsilon}{4}, \quad (4.27)$$

se cumplan.

Si hacemos $A = \ln(n^p)$ con $p > 0$, existe un número natural $N_A(p)$ tal que si $n > N_A(p)$, entonces $A = \ln(n^p) > A_2$.

La ecuación (4.24) se transforma en:

$$\frac{4n^{-p\gamma}}{1 - n^{-p\omega} - 4n^{-p\gamma}}(L(b-a) + |f(b)|) < \frac{\varepsilon}{4},$$

y existe un número natural $N_1(p)$ tal que si $n > N_1(p)$, esta desigualdad se cumple.

La ecuación (4.25) se puede expresar como

$$\omega p L(b-a) \frac{\ln(n)}{n} < \varepsilon.$$

Por (4.21) existe un número natural $N_2(p)$ tal que si $n > N_2(p)$, esta desigualdad se cumple.

La ecuación (4.26) se transforma en:

$$\omega(L(b-a) + |f(b)|) p \ln(n) s(\gamma p \ln(n)) (1 - s(\gamma p \ln(n))) < \frac{\varepsilon}{4}.$$

Por (4.19) existe un número natural $N_3(p)$ tal que si $n > N_3(p)$, esta desigualdad se cumple.

Finalmente, existe un número natural N_4 tal que si $n > N_4$, (4.27) se satisface.

El resultado queda probado si tomamos

$$N = \max(N_A(p), N_1(p), N_2(p), N_3(p), N_4).$$

□

Teorema 3 Sea f una función continua en $[a, b]$. Para cada $\varepsilon > 0$ existe una función $A(n)$ y un número natural N tal que

$$|f(x) - N_a(x, A(n))| < \varepsilon$$

para todo $n > N$ y para todo $x \in [a, b]$.

Demostración. Utilizando el Lema 6, podemos encontrar una función lineal a trozos $l_{\frac{\varepsilon}{2}}$ tal que

$$|f(x) - l_{\frac{\varepsilon}{2}}(x)| < \frac{\varepsilon}{2} \text{ para todo } x \in [a, b].$$

Como $l_{\frac{\varepsilon}{2}}$ es Lipschitziana en $[a, b]$, podemos aplicar el Lema 8 y encontrar $A(n)$ y N tal que

$$|l_{\frac{\varepsilon}{2}}(x) - N_a(x, A(n))| < \frac{\varepsilon}{2} \text{ para todo } x \in [a, b] \text{ y } n > N.$$

El resultado se deduce de las desigualdades anteriores.

□

Las Figuras 4.2 y 4.3 muestran la convergencia uniforme de las ai-nets cuando n y A se incrementan. Aquí hemos considerado $\alpha = 1$, $\beta = -1$ y $p = 3$.

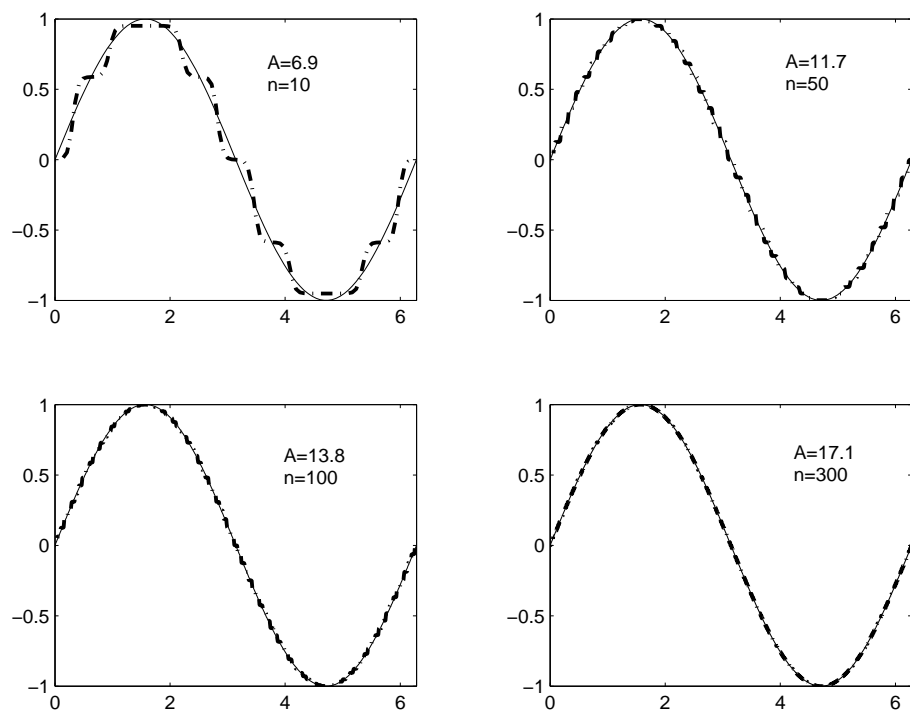


Figura 4.2: Convergencia uniforme de ai-nets: $\sin(x)$.

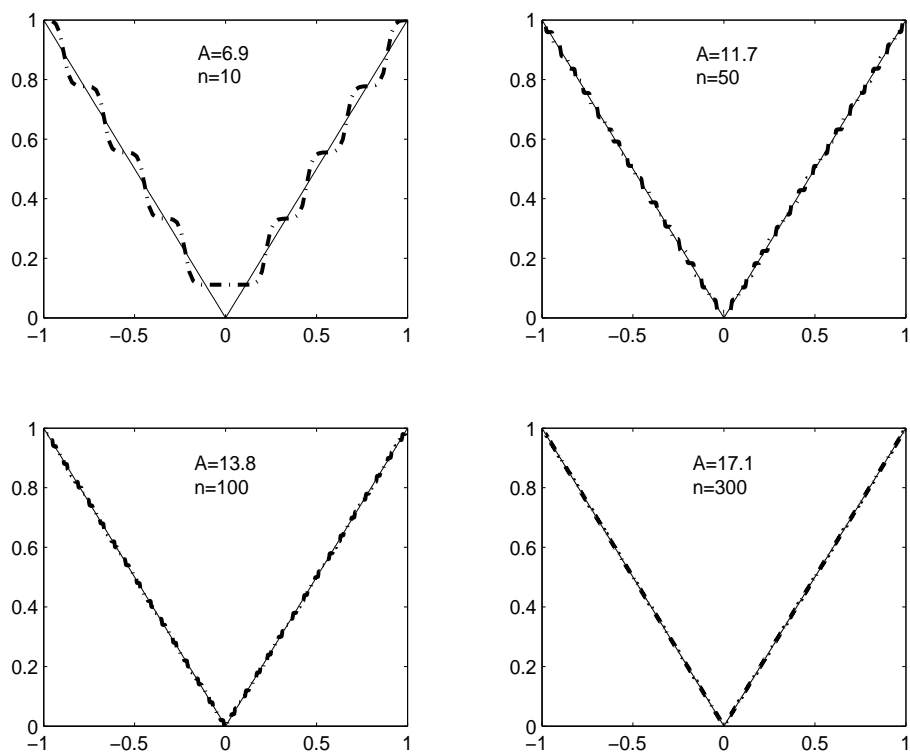


Figura 4.3: Convergencia uniforme de ai-nets: $|x|$.

El Teorema 3 se puede extender al caso multidimensional de acuerdo a los procedimientos dados en (Chen, Chen y Liu, 1991; Hornik, Stinchcombe y White, 1989). Nosotros seguimos la primera aproximación, menos general pero más constructiva.

4.5. Una demostración del Teorema de Cybenko

Corolario 3 *Sea $f(\mathbf{x})$ una función continua en $I = [0, 1]^d$. Para cada $\varepsilon > 0$ existe una red neuronal*

$$N(\mathbf{x}) = \sum_{j=0}^n c_j s(\mathbf{w}_j \cdot \mathbf{x} + b_j)$$

donde $\mathbf{w}_j \in R^d$, $c_j, b_j \in R$ tal que

$$|N(\mathbf{x}) - f(\mathbf{x})| < \varepsilon \text{ para todo } \mathbf{x} \in I.$$

Demostración. Se dan los pasos principales de la demostración. Más detalles se pueden consultar en (Chen, Chen y Liu, 1991).

- Se extiende $f(\mathbf{x})$ para que sea una función 2-periódica par con respecto a cada variable x_i , $i = 1, \dots, d$, $g(\mathbf{x})$ definida en $J = [-1, 1]^d$.
- Usando un resultado sobre la media de Bochner-Riesz (Stein y Weiss, 1975)[p.256]) y la paridad de $g(\mathbf{x})$, se tiene que para cualquier $\varepsilon > 0$, existe un $R > 0$ tal que para cualquier $\mathbf{x} = (x_1, \dots, x_d) \in J$

$$\left| \sum_{|\mathbf{m}| \leq R} c_{m_1 \dots m_d} \cos(m_1 x_1 + \dots + m_d x_d) - g(x_1, \dots, x_d) \right| < \frac{\varepsilon}{2},$$

donde $c_{m_1 \dots m_d}$ son números reales, $\mathbf{m} = (m_1, \dots, m_d) \in \mathbb{Z}^d$ y $|\mathbf{m}|^2 = m_1^2 + \dots + m_d^2$.

- Se define $u = \mathbf{m} \cdot \mathbf{x}$ y se aproxima $\cos(u)$ por una ai-net.

□

4.6. Aproximación uniforme de una función afín por medio de dos neuronas

Attali y Pagés (1997) probaron el siguiente:

Teorema Un polinomio de grado 1 y d variables, puede ser aproximado uniformemente sobre subconjuntos compactos de \mathbb{R}^d por una SLFN con $d + 1$ neuronas, suponiendo que la función de activación es diferenciable y que su valor y el de su primera derivada es diferente de cero en el origen.

Aunque la demostración es semiconstructiva, Attali y G.Pagés no proporcionan una expresión explícita de los pesos de la red en función del error uniforme.

Malakooti y Yingquing (1998) consideran SLFNs con función de activación logística y demuestran un teorema similar al anterior, pero reduciendo a 2 el

número de neuronas necesarias y considerando el caso de dominios de la forma: $\{x \in \mathbb{R}^d / |\mathbf{a} \cdot \mathbf{x} + a_0| < M\}$, donde $f(x) = \mathbf{a} \cdot \mathbf{x} + a_0$ es la función afín a aproximar y M es un número positivo arbitrario.

La demostración es constructiva y los autores presentan un algoritmo para obtener los pesos de la red, que proporciona valores muy pequeños para los pesos interiores y valores enormes para los pesos exteriores, incluso en el caso de errores uniformes relativamente grandes. De hecho, los pesos internos disminuyen como $1/M^2$. Esto implica falta de precisión e inestabilidad en la red resultante cuando se desean obtener buenos aproximantes en regiones arbitrarias o cuando los coeficientes de la función afín se conocen con un cierto margen de error.

En esta sección se extiende el resultado de Malakooti y Zhou a regiones arbitrarias en forma de politopo. El algoritmo resultante permite obtener unos pesos internos que disminuyen como $1/m$, siendo m el valor mínimo de $|\mathbf{u} \cdot \mathbf{x}|$ sobre los puntos del politopo y \mathbf{u} el cociente de \mathbf{a} por su módulo. Como en general m y M son de igual orden de magnitud, el método propuesto es más estable y puede admitir valores de la precisión mayores que los considerados por el algoritmo de Malakooti y Zhou.

Se ha definido formalmente una SLFN en 2.2.1.

Sea $\{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d\}$ un conjunto de $d + 1$ vectores en \mathbb{R}^d . Este conjunto se denomina independientemente afín si los vectores $\{\mathbf{v}_1 - \mathbf{v}_0, \mathbf{v}_2 - \mathbf{v}_0, \dots, \mathbf{v}_d - \mathbf{v}_0\}$ son linealmente independientes. Supongamos ahora que $\{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d\}$ es un subconjunto independientemente afín de \mathbb{R}^d . El d -simplex T generado por $\{\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d\}$, especificado por $\langle \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d \rangle$, se define como la envolvente

convexa de los vectores $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d$. Denotamos por $|T|$ el diámetro de un d -simplex T .

A continuación se demuestra que una función afín definida sobre un d -simplex en \mathbb{R}^d puede ser aproximada uniformemente por una red neuronal de una capa que tiene sólo dos neuronas y cuyos pesos se pueden expresar de forma analítica cerrada. Para simplificar la explicación, asumiremos que la función de activación es la función logística.

Usando la forma de Lagrange del resto en el desarrollo limitado de Taylor, en el método para aproximar polinomios por redes neuronales dado en (Attali y Pagés, 1997), tenemos:

Proposición 1 Sea $p(x) = A_0 + A_1x$ un polinomio de grado 1, en una variable y con coeficientes reales. Dado $\varepsilon > 0$, existe una SLFN con dos neuronas

$$N(x) = \sum_{j=0}^1 \lambda_j s(\alpha_j x)$$

tal que

$$|N(x) - p(x)| < \varepsilon, \quad x \in [0, 1],$$

donde $s(x) = 1/(1 + e^{-x})$ (función logística) y $\lambda_j, \alpha_j, j = 0, 1$, son funciones de A_0, A_1 y ε que pueden ser expresadas en una forma analítica cerrada.

Demostración. Sea $\alpha > 0$. Aplicando el teorema de Taylor con el resto de Lagrange a la función $s(\alpha x)$, podemos escribir

$$s(\alpha x) = \frac{1}{2} + \frac{\alpha x}{4} + \frac{\alpha^2 x^2}{2} s''(z), \quad z \in (0, \alpha x).$$

Sean $\alpha_0, \alpha_1, \lambda_0, \lambda_1$ variables positivas. Si reemplazamos sucesivamente en la ecuación anterior, α por α_0 y α_1 , multiplicamos la ecuación resultante por λ_0 y λ_1 respectivamente, y las sumamos, tenemos

$$r \equiv \left| \sum_{j=0}^1 \lambda_j s(\alpha_j x) - \frac{1}{2} \sum_{j=0}^1 \lambda_j - \frac{x}{4} \sum_{j=0}^1 \lambda_j \alpha_j \right| = \frac{x^2}{2} \left| \sum_{j=0}^1 \lambda_j \alpha_j^2 s''(z_j) \right|,$$

donde $z_j \in (0, \alpha_j x)$, $j = 0, 1$. Dado que $s'' = 2s^3 - 3s^2 + s$ y $s(z) \in (0, 1)$ se tiene que $|s''(z)| < 1/10$ para todo $z \in \mathbb{R}$. Por ello, ya que $x \in [0, 1]$, se tiene:

$$r < \frac{1}{20} \left| \sum_{j=0}^1 \lambda_j \alpha_j^2 \right|. \quad (4.28)$$

Si consideramos el sistema

$$\sum_{j=0}^1 \lambda_j = 2A_0,$$

$$\sum_{j=0}^1 \lambda_j \alpha_j = 4A_1,$$

y hacemos

$$\alpha_j = c_j h, \quad (4.29)$$

donde $c_j > 0$, $j = 0, 1$, y $h > 0$, tenemos:

$$\lambda_0 = (2A_0c_1h - 4A_1)/(c_1 - c_0)h, \quad (4.30)$$

$$\lambda_1 = (4A_1 - 2A_0c_0h)/(c_1 - c_0)h, \quad (4.31)$$

De (4.28-4.31) se sigue

$$r < h(|2A_0c_1h - 4A_1|c_0^2 + |4A_1 - 2A_0c_0h|c_1^2)/(20|c_1 - c_0|). \quad (4.32)$$

De (4.32), usando la identidad:

$$\max_{|h| \leq 1} |\gamma h + \delta| = \max(|\gamma + \delta|, |\gamma - \delta|), \quad \gamma, \delta \in \mathbb{R}$$

y definiendo

$$M1 \equiv c_0^2 \max(|2A_0c_1 - 4A_1|, |2A_0c_1 + 4A_1|),$$

$$M2 \equiv c_1^2 \max(|4A_1 - 4A_0c_0|, |4A_1 + 2A_0c_0|,$$

$$M \equiv (M1 + M2)/(20|c_1 - c_0|),$$

se sigue que si $h \leq 1$, entonces $r < Mh$. Por tanto, dado $\varepsilon > 0$, es suficiente hacer $h = \min(1, \varepsilon/M)$. Los pesos de la red están dados por (4.29), (4.30) y (4.31).

□

Nota 1. La proposición 1 es válida para cualquier función de activación ϕ que satisfaga las siguientes condiciones:

1. $\phi \in C^2(\mathbb{R}, \mathbb{R})$,
2. $\phi(0) \neq 0, \phi'(0) \neq 0$,
3. $|\phi''(z)|$ está acotada en \mathbb{R} .

Nota 2. El sistema lineal que aparece en la demostración de la proposición 1

$$\begin{pmatrix} 1 & 1 \\ c_0 h & c_1 h \end{pmatrix} \begin{pmatrix} \lambda_0 \\ \lambda_1 \end{pmatrix} = \begin{pmatrix} 2A_0 \\ 4A_1 \end{pmatrix},$$

puede estar mal condicionado. Este hecho no implica la inestabilidad de la SLFN resultante, porque depende de los pesos en la siguiente forma

$$N(x) = (\lambda_0 + \lambda_1)/2 + (c_0 h \lambda_0 + c_1 h \lambda_1)x/4.$$

Corolario 4 Sea $p^*(t) = a_0 + a_1 t$ un polinomio de grado 1, de una variable y con coeficientes reales. Dado $\varepsilon > 0$ y dos números arbitrarios reales t_l y t_u tales que $t_l < t_u$, existe una SLFN con dos neuronas

$$N^*(t) = \sum_{j=0}^1 \lambda_j^* s(\alpha_j^* t + \beta_j^*),$$

tal que

$$|N^*(t) - p^*(t)| < \varepsilon. \quad t \in [t_l, t_u],$$

donde $\lambda_j^*, \alpha_j^*, \beta_j^*$, $j = 0, 1$, son funciones de a_0, a_1 y ε que pueden ser expresadas de forma analítica cerrada.

Demostración. Si hacemos el cambio de variable

$$t = (t_u - t_l)x + t_l,$$

y definimos

$$A_0 \equiv a_0 + a_l t_l,$$

$$A_1 \equiv a_l(t_u - t_l),$$

el problema es equivalente a aproximar $p(x) = A_0 + A_1 x$ sobre $[0, 1]$. Sea $N(x)$ la red neuronal dada por la proposición 1, esto es, tal que

$$|N(x) - p(x)| < \varepsilon, \quad x \in [0, 1],$$

Ya que $x = (t - t_l)/(t_u - t_l)$

$$|N(x) - p(x)| = |N((t - t_l)/(t_u - t_l)) - a_0 - a_l t| < \varepsilon, \quad t \in [t_l, t_u].$$

Ya que $N(x) = \sum_{j=0}^1 \lambda_j s(\alpha_j x)$, se sigue que

$$N^*(t) = \sum_{j=0}^1 \lambda_j s((\alpha_j t - \alpha_j t_j)/(t_u - t_l)).$$

□

El principal resultado de esta sección es el siguiente:

Teorema 4 Sea $a(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x} + a_0$ una aplicación afín definida sobre el d – simplex $T = \langle \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d \rangle$. Dado $\varepsilon > 0$, existe una SLFN con dos neuronas

$$N(\mathbf{x}) = \sum_{j=0}^1 c_j s(\mathbf{w}_j \mathbf{x} + b_j),$$

tal que

$$|N(\mathbf{x}) - a(\mathbf{x})| < \varepsilon, \quad \mathbf{x} \in T, \quad (4.33)$$

donde $c_j, b_j \in \mathbb{R}$ y $\mathbf{w}_j \in \mathbb{R}^d$ son funciones de \mathbf{a}, a_0 y ε que pueden ser expresadas en una forma analítica cerrada.

Demostración. Sea $\mathbf{u} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$. Consideremos la línea recta $\mathbf{x} = t\mathbf{u}$. La función afín sobre esta línea recta es

$$h(t) = a(\mathbf{x}(t)) = t\mathbf{a} \cdot \mathbf{u} + a_0.$$

Si definimos

$$t_u = \max_{i=0,\dots,d} \{\mathbf{u} \cdot \mathbf{v}_i\},$$

$$t_l = \min_{i=0,\dots,d} \{\mathbf{u} \cdot \mathbf{v}_i\},$$

$h(t)$ es un polinomio de grado 1 definido sobre $[t_l, t_u]$, por tanto, por el Corolario 4, existe una red $N^*(t)$ satisfaciendo

$$|N^*(t) - h(t)| < \varepsilon \quad \text{para todo } t \in [t_l, t_u] \quad (4.34)$$

Definimos

$$N(\mathbf{x}) \equiv N^*(\mathbf{u} \cdot \mathbf{x}).$$

Para probar (4.33) es suficiente considerar que cualquier $\mathbf{x} \in T$ puede ser escrito como

$$\mathbf{x} = \mathbf{x}_0 + t\mathbf{u},$$

donde \mathbf{x}_0 es la proyección de \mathbf{x} sobre el subespacio afín que contiene el origen y es ortogonal a \mathbf{u} , por tanto, $\mathbf{x}_0 \cdot \mathbf{u} = 0$ y $t \in [t_l, t_u]$. Dado que

$$N(\mathbf{x}) = N^*(\mathbf{u} \cdot (\mathbf{x}_0 + t\mathbf{u})) = N^*(t),$$

$$a(\mathbf{x}) = \mathbf{a} \cdot (\mathbf{x}_0 + t\mathbf{u}) + a_0 = t\mathbf{a} \cdot \mathbf{u} + a_0 = h(t),$$

el resultado se sigue de (4.34).

□

Capítulo 5

Un método indirecto para la construcción de árboles neuronales de aprendizaje rápido. Aplicación a la representación de objetos

5.1. Introducción. Necesidad de Aprendizaje rápido.

Hemos abordado en el capítulo 4, el problema de la interpolación y aproximación de funciones mediante redes neuronales. Nuestro problema consiste en representar la función distancia a un objeto por medio de una red neuronal. Los

datos que entran en el córtex parietal son ejemplos suministrados principalmente por el sistema visual, como hemos visto en el capítulo 1, y los procedentes de la situación del propio cuerpo, en nuestro caso concreto la mano, para proceder a la ejecución de un movimiento de aproximación de dicha mano al objeto. Esos datos constituyen una función que nuestro sistema tiene que aproximar de forma automática y rápida. Las redes neuronales que sirven para realizar dicha aproximación aprenden de los ejemplos y deben ser capaces de representar el objeto para actuar sobre él, lo que constituye una función prototípica de la vía dorsal.

Los algoritmos de aprendizaje tradicional usados para entrenar redes neuronales «feedforward» se basan en metodologías de búsqueda iterativa, e.g. retro-propagación (BP) y sus variantes. Estos algoritmos de entrenamiento consumen mucho tiempo, especialmente para aplicaciones con conjuntos de datos a gran escala y/o una gran expectativa de precisión, como son los requeridos para nuestro problema. La necesidad de algoritmos de aprendizaje rápidos para redes neuronales ha sido enfatizada por algunos autores (Huang, 2003; Huang, Zhu y Siew, 2006a; Huang, Zhu y Siew, 2006b). El aprendizaje rápido es necesario en aplicaciones donde el tiempo es crítico y el entorno cambia rápidamente, donde el conocimiento tiene que construirse consecutivamente. Esto ocurre en el caso del procesamiento de las entradas sensoriales que ocurre en el córtex parietal, el conjunto de puntos de entrenamiento no está dado a priori y se genera un flujo continuo de datos suministrado por el sistema sensorial. Por otro lado, la ejecución de las redes neuronales para realizar una función, una vez que han aprendido, ha de ser rápida y precisa, como es el caso del tiempo que se necesita para calcular la distancia a un objeto. En este capítulo, cumpliendo los requisitos metodológi-

cos que hemos visto en el capítulo 3 se presenta una arquitectura autogenerada mediante árboles neuronales y un algoritmo que cumple esas demandas.

Se presenta un método indirecto de aprendizaje que satisface las condiciones A, B y C que se han definido en la sección 2.5. Los resultados experimentales confirman propiedades de generalización (aproximación) excelentes sobre problemas prácticos reales, (Condición D) (de 2.5). El capítulo está organizado como sigue: La sección 5.2 muestra el procedimiento para obtener el aproximante matemático. Consiste en una función afín por trozos. Esta función se obtiene por un método de división adaptativa. El criterio de división se obtiene utilizando la fórmula de cubatura de Grundmann-Möller. Llamamos ASBC al algoritmo resultante. La sección 5.3 describe el método para generar el árbol de aproximación. En la sección 5.4 se estudia la estructura global de la red obtenida a partir del árbol de aproximación. Se detalla la estructura de los nodos no-hoja. El algoritmo que consiste en aplicar ASBC y después reemplazar las funciones afines que componen la aproximación por redes SLFN con dos neuronas en la capa oculta, se denomina NTASBC. La sección 5.5 estudia la aplicación de NTASBC al problema de aprendizaje de la función distancia de un punto variable a cuerpos convexos y no convexos en dos y tres dimensiones. La sección 5.6 compara el rendimiento de los árboles generados por NTASBC con redes SLFN entrenadas por un método de regiones de confianza basado en el algoritmo de Newton (interior-reflexivo), un método de optimización (retropropagación) muy eficiente. También se aportan tiempos de test individuales del algoritmo exacto y las redes consideradas. La sección 5.7 resume las conclusiones.

5.2. Aproximación por trozos de funciones por el método de división adaptativa basado en cubatura (ASBC).

Sea $\Pi_1(\mathbb{R}^d)$ el espacio de polinomios de grado menor o igual a 1, esto es

$$\Pi_1(\mathbb{R}^d) = \{a_0 + a_1x_1 + \dots + a_dx_d : a_i \in \mathbb{R}, i = 0, \dots, d\}.$$

Consideremos el d -intervalo $I = [\mathbf{b}, \mathbf{c}]$, donde $\mathbf{b}, \mathbf{c} \in \mathbb{R}^d$. Podemos encontrar n símlices (cerrados) T_i tal que

$$\begin{aligned} \mathring{T}_i \cap \mathring{T}_j &= \emptyset, \quad i \neq j, \\ \bigcup_{i=1}^n T_i &= I, \end{aligned}$$

donde \mathring{T} es el interior del conjunto T (por ejemplo, ver (Bliss y Su, 2005), (Orden y Santos, 2003). Denotamos por $\Delta \equiv \{T_i\}_{i=1}^n$ una partición de I . El conjunto de particiones de I se denota por $\mathcal{P}(I)$. Se tiene el siguiente resultado:

Proposición 2. (Gamelin and Greene, 1999). Sea $T = \langle \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d \rangle$ un d -simplex y f una función continua de T a \mathbb{R} , entonces existe una única correspondencia afín $a_T(f)$ tal que

$$a_T(f)(\mathbf{v}_j) = f(\mathbf{v}_j), \quad j = 0, 1, \dots, d.$$

Sea S un conjunto compacto en \mathbb{R}^d , sea $C(S)$ la clase de funciones g con dominio S y rango en \mathbb{R} tal que g es continua sobre S . La norma habitual para

$C(S)$ está dada por

$$\|g\|_{C(S)} = \max_{\mathbf{x} \in S} |g(\mathbf{x})|.$$

Del anterior resultado, dado $f \in C(T)$ podemos definir el operador de aproximación local

$$\begin{aligned} a_T : C(T) &\rightarrow \Pi_1(\mathbb{R}^d) \\ f &\rightarrow a_T(f). \end{aligned}$$

La función afín $a_T(f)$ tiene las siguientes propiedades:

Proposición 3. Sea $T = \langle \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d \rangle$ un d -simplex contenido en I y $f \in C(I)$, entonces

- a) Si $\mathbf{x} = \sum_{i=0}^d \lambda_i \mathbf{v}_i$, $\lambda_i \geq 0$, $i = 0, \dots, d$ y $\sum_{i=0}^d \lambda_i = 1$, entonces

$$a_T(f)(\mathbf{x}) \equiv \sum_{i=0}^d \lambda_i f(\mathbf{v}_i).$$

- b) Dado $\varepsilon > 0$ existe un número $\delta > 0$ tal que para todos los d -simplices T contenidos en I con $|T| < \delta$ se cumple que $\|f - a_T(f)\|_{C(T)} < \varepsilon$.

Demostración. a) Se deduce de la Proposición 2. b) Es una consecuencia de a) y de la continuidad uniforme de f sobre I .

Sea Δ una partición de I y $f \in C(I)$ definimos el aproximante

$$s = \sum_{T_i \in \Delta} a_{T_i}(f) \chi_{T_i},$$

donde χ_{T_i} es la función característica del conjunto T_i . Se puede probar que $s \in C(I)$.

Intentamos resolver el siguiente problema: Dado $\varepsilon > 0$ y $f \in C(I)$, encontrar $\Delta \in \mathcal{P}(I)$ tal que

$$\|f - s\|_{C(I)} = \max_{T_i \in \Delta} \|f - a_{T_i}(f)\|_{C(T_i)} < \varepsilon.$$

Un procedimiento para resolver este problema consiste en usar algoritmos de división adaptativa (o aproximación por árboles). Estos algoritmos consideran:

- Una función no negativa $U(T)$, la cual da una estimación o cota superior de $\|f - a_T(f)\|_{C(T)}$.
- Una partición inicial Δ_1 de I .
- Un algoritmo R para dividir cualquier simplex en dos o más simples.

Dado $\varepsilon > 0$ decimos que un simplex T es **bueno** si $U(T) < \varepsilon$; en caso contrario lo llamamos **malo**.

El algoritmo procede de la siguiente forma:

- Los simples buenos en la partición inicial Δ_1 se sitúan en el conjunto \mathcal{G}_1 y los simples malos se sitúan en el conjunto \mathcal{B}_1 .
- En cada paso j tenemos un conjunto de simples buenos \mathcal{G}_j y un conjunto de simples malos \mathcal{B}_j . Dividimos los simples en \mathcal{B}_j en subsimplices de

acuerdo a R y estudiamos si estos subsímplices T son buenos o malos para obtener los conjuntos \mathcal{G}_{j+1} y \mathcal{B}_{j+1} .

- El algoritmo se detiene si $\mathcal{B}_j = \emptyset$. Entonces \mathcal{G}_j es la partición buscada Δ . Si el criterio de parada no se satisface, se repite el proceso.

Este algoritmo siempre converge debido a la proposición 3.b.

La aproximación polinomial a trozos por división adaptativa fue introducida por Birman y Solomiak (1967) en el contexto de los espacios de Sobolev y polinomios de grado n . Este método ha sido estudiado desde un punto de vista teórico en, por ejemplo, Binev y De Vore (2004), Birman y Solomiak (1967), de Boor y Rice (1979), Dahmen (1982), De Vore y Yu (1990). En la práctica, la mayor dificultad consiste en calcular el valor de

$$E(T) = \max_{\mathbf{x} \in T} |f(\mathbf{x}) - a_T(f)(\mathbf{x})|. \quad (5.2)$$

Recientemente Karavaivanov, Petrushev y Sharpely (2003), han propuesto un algoritmo («Push the error») para computar (5.2). Este algoritmo está limitado al caso $d = 2$. Por tanto, la aproximación de funciones de más de dos variables requiere la introducción de nuevas técnicas. En este capítulo, usamos la alternativa a (5.2)

$$\bar{E}(T) \equiv \frac{\int_T |f(\mathbf{x}) - a_T(f)(\mathbf{x})| d\mathbf{x}}{v(T)}, \quad (5.3)$$

donde $v(T)$ denota el volumen del d -simplex T .

Mientras (5.2) garantiza conseguir un aproximante uniforme, (5.3) proporciona un aproximante cuya precisión debe ser evaluada empíricamente.

Para calcular la integral de la parte derecha de (5.3), hemos usado la fórmula de Grundmann y Möller (1978).

Sea $S = \langle (0, 0, \dots, 0), (1, 0, \dots, 0), (0, 1, \dots, 0), \dots, (0, 0, \dots, 1) \rangle$ el d -simplex standard en \mathbb{R}^d . Si $p = 2s + 1$ y $s \in \mathbb{Z}^+ \cup \{0\}$, la regla de cubatura de grado p está dada por

$$\int_S g d\mathbf{x} \approx \sum_{i=0}^s (-1)^i 2^{-2s} \frac{(p+d-2i)^p}{i!(p+d-i)!} \sum_{\substack{|\beta|=s-i \\ \beta_0 \geq \dots \geq \beta_d}} g \left(\left(\frac{2\beta_0+1}{p+d-2i}, \dots, \frac{2\beta_d+1}{p+d-2i} \right)_p \right), \quad (5.4)$$

donde $\beta \equiv (\beta_0, \beta_1, \dots, \beta_d)$ es una $(d+1)$ -tupla de números enteros no negativos con $|\beta| \equiv \beta_0 + \beta_1 + \dots + \beta_d$. $(\mathbf{y})_p \equiv (y_0, y_1, \dots, y_d)_p$ denota el conjunto de todas las d -tuplas que son las últimas d componentes de todas las $(d+1)$ -tuplas derivadas de \mathbf{y} por cualquier permutación de las $d+1$ componentes de \mathbf{y} . El número de puntos de integración en (5.4) es $\binom{d+s+1}{s}$. Esta fórmula es exacta para polinomios de grado p .

Cualquier d -simplex $T = \langle \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d \rangle$ es la imagen de S bajo una transformación afín ϕ

$$T = \phi(S).$$

Usando la fórmula de cambio de variables podemos calcular la integral de una función h sobre T como una integral sobre S

$$\int_T h(\mathbf{y}) d\mathbf{y} = \int_{\phi(S)} h(\mathbf{y}) d\mathbf{y} = \int_S h(\phi(\mathbf{x})) |J| d\mathbf{x},$$

donde $|J|$ es el valor absoluto del determinante Jacobiano de ϕ . Dado que ϕ es afín tenemos

$$\bar{E}(T) = d! \int_S |f(\phi(\mathbf{x})) - a_T(f)(\phi(\mathbf{x}))| d\mathbf{x}. \quad (5.5)$$

La parte derecha de (5.5) se calcula usando (5.4).

Los cálculos se han hecho usando las reglas de Grundmann y Möller de grado $p = 5, 7, 9, 11, 13$, y 15 con $d = 2$ y $d = 3$. Debido a la carencia de estimaciones de error para las reglas de Grundmann y Möller, hemos seguido la práctica habitual de comparar valores sucesivos de la integral correspondientes a incrementar el grado, tomando el valor obtenido cuando la sucesión se vuelve estacionaria. Denominamos $\bar{E}_p(T)$ al valor de (5.5) calculado con la regla de cubatura de grado p . El procedimiento propuesto se describe con el siguiente pseudocódigo (tipo lenguaje C).

```

Hacer  $p = 5$ ;
Calcular  $\bar{E}_5(T)$  and  $\bar{E}_7(T)$ ;
while ( $|\bar{E}_p(T) - \bar{E}_{p+2}(T)| \geq \varepsilon/10$  &&  $p < 13$ )
{
     $p = p + 2$ ;
}
if ( $|\bar{E}_p(T) - \bar{E}_{p+2}(T)| < \varepsilon/10$ )     $\bar{E}(T) = \bar{E}_{p+2}(T)$ ;
else                                      $\bar{E}(T) = 10^6$ ;

```

En muchos algoritmos adaptativos, la estimación del error local ERR se toma simplemente como el valor absoluto de la diferencia entre dos aproximaciones por cubatura (Berntsen y Espelid, 1991). De esta forma, tenemos la acotación aproximada

$$|I - \hat{I}| < ERR,$$

donde I es el valor exacto de la integral e \hat{I} es el valor calculado por la regla de cubatura. Cuando $\hat{I} \simeq \varepsilon$, parece natural considerar que $ERR \ll \varepsilon$. En el pseudocódigo anterior, hemos tomado $ERR = \varepsilon/10$.

5.3. Algoritmo para generar el árbol de aproximación

En esta sección se presenta un pseudocódigo (tipo lenguaje C) del algoritmo utilizado para generar el árbol de aproximación. Suponemos que el dominio inicial \mathcal{D} es la unión de varios d -simplices con interiores disjuntos dos a dos. Aplicamos el algoritmo a cada uno de ellos. Empezamos por definir una estructura de tipo nodo

```
struct node
{
double  $\mathbf{v}_0[d]$ ;
double  $\mathbf{v}_1[d]$ ;
...;
double  $\mathbf{v}_d[d]$ ;
int  $spc$ ;
int  $lfn$ ;
int  $f$ ;
int  $l$ ;
int  $r$ ;
}N[ ],
```

donde, \mathbf{v}_i , $i = 0, \dots, d$ son los vértices del simplex considerado, y

spc : variable que toma el valor 1 si el algoritmo ha aplicado el *criterio de división* al simplex y 0 en caso contrario.

lfn : variable que toma el valor 1 si el nodo no tiene hijos (*nodo hoja*) y 0 en caso contrario.

f : número del nodo padre del nodo actual.

l : hijo izquierdo del nodo actual (-1 en el caso de los nodos hoja).

r : hijo derecho del nodo actual (-1 en el caso de los nodos hoja).

Dados dos parámetros positivos ε y ML , el *criterio de división* se define como

$$SPC(T) \equiv \{\bar{E}(T) > \varepsilon \text{ y } MAXL > ML\},$$

donde $\bar{E}(T)$ es la aproximación a $E(T)$ descrita en la sección anterior y $MAXL$ denota la longitud de la arista más grande del simplex. Imponemos un valor mínimo ML para la longitud de las aristas de los subsimplices. En los experimentos (sección 5.5), ML es un número muy pequeño y no ha sido nunca alcanzado.

Ahora introducimos la notación

$$\langle \mathbf{v} \rangle \equiv \langle \mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_d \rangle .$$

Algoritmo para generar el árbol de aproximación

- **Entrada de datos:** $\mathbf{v0}_i$, $i = 0, 1, \dots, d$ (vértices del d -simplex inicial)
 $\varepsilon, ML, po = 0$.

- **Inicializar** el nodo 0

$$N[0].\mathbf{v} = \mathbf{v0}; \quad N[0].spc = 0; \quad N[0].f = -1;$$

- **Calcular** la longitud máxima ($MAXL$) de las aristas de $T_0 = \langle \mathbf{v0} \rangle$.

- **if** ($SPC(T_0) == 1$) {dividir la arista más larga de T_0 y obtener dos subsimplices $T_1 = \langle \mathbf{v1} \rangle$ and $T_2 = \langle \mathbf{v2} \rangle$ } **else** {exit }

- Inicializar los nodos 1 y 2

$$N[1].\mathbf{v} = \mathbf{v1}; \quad N[1].f = 0; \quad N[1].spc = 0 \quad N[1].r = -1;$$

$$N[2].\mathbf{v} = \mathbf{v2}; \quad N[2].f = 0; \quad N[2].spc = 0 \quad N[2].r = -1;$$

- Completar los miembros del nodo 0

$$N[0].spc = 1; \quad N[0].lfn = 0; \quad N[0].l = 1; \quad N[0].r = 2;$$

- Determinar el nodo actual y el contador de nodos

$$c = 1; \quad nn = 2;$$

- **while**($po \neq 2$)

{

$$rc = N[c].r;$$

$$\mathbf{if}(N[c].spc == 0)$$

{

- **Calcular** la longitud máxima ($MAXL$) de las aristas de $T_c = \langle N[c].\mathbf{v} \rangle$.

if($SPC(T_c) == 1$)

{

- Dividir la arista más larga de T_c y obtener dos subsimplices

$T_1 = \langle \mathbf{v1} \rangle$ y $T_2 = \langle \mathbf{v2} \rangle$;

Inicializar los nodos $nn + 1$ y $nn + 2$

$N[nn + 1].\mathbf{v} = \mathbf{v1}$; $N[nn + 1].f = c$;

$N[nn + 1].spc = 0$; $N[nn + 1].r = -1$;

$N[nn + 2].\mathbf{v} = \mathbf{v2}$; $N[nn + 2].f = c$;

$N[nn + 2].spc = 0$; $N[nn + 2].r = -1$;

- Completar los miembros del nodo actual

$N[c].spc = 1$; $N[c].lfn = 0$; $N[c].l = nn + 1$; $N[c].r = nn + 2$;

$c = nn + 1$; (nuevo nodo actual)

$nn = nn + 2$;

}

else

{

- Definir el nodo como un nodo hoja

$N[c].spc = 1$; $N[c].lfn = 1$; $N[c].l = -1$; $N[c].r = -1$;

- Determinar el nuevo nodo actual

```

         $c = N[c].f;$ 

        if( $c == 0$ )  $po = po + 1;$ 
    }
}

else if(  $N[rc].spc == 0$  )
{
     $c = rc;$ 

    if ( $c == 0$ )  $po = po + 1;$ 
}

else if(  $N[rc].spc == 1$  )
{
     $c = N[c].f;$ 

    if ( $c == 0$ )  $po = po + 1;$ 
}
}

```

po es una variable que cuenta el número de pasos del flujo del programa a través del nodo raíz. La generación del árbol de aproximación sigue el método de preorden, como se puede ver en la Figura 5.1.

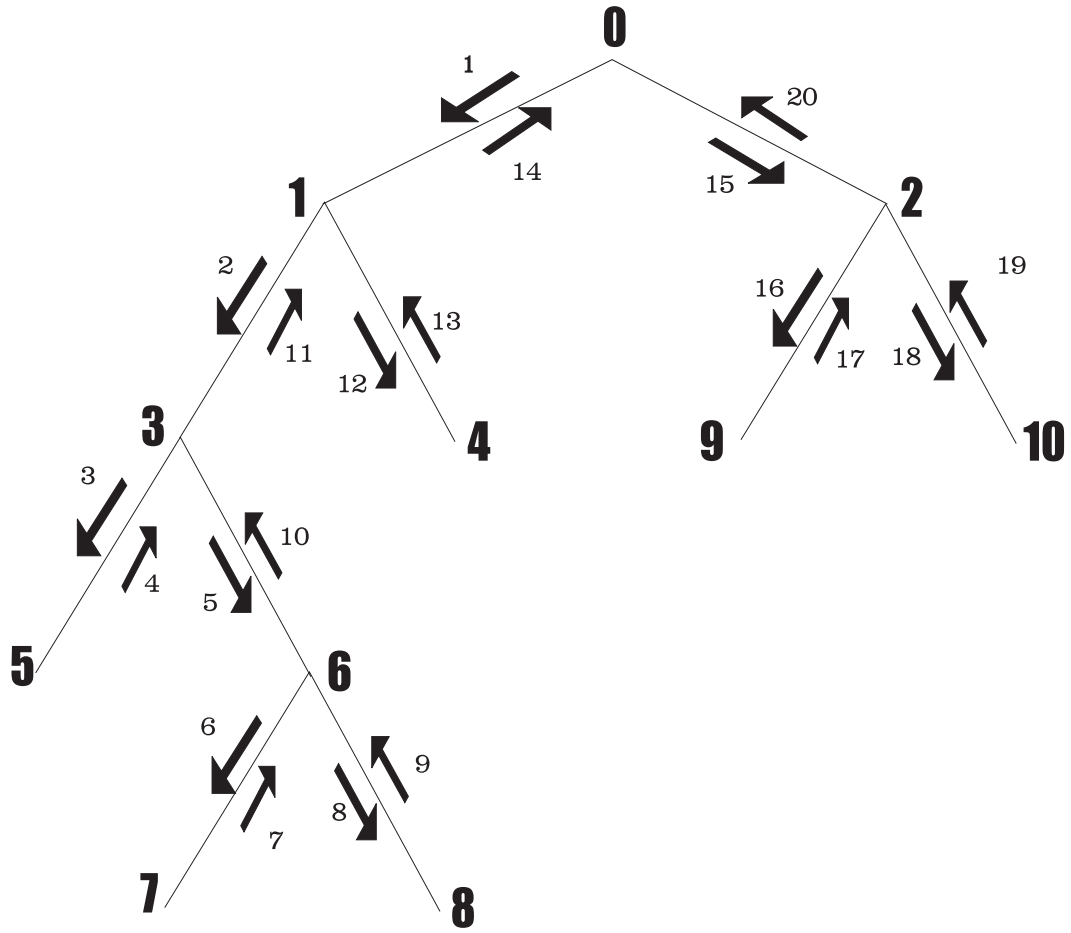


Figura 5.1: Generación del árbol de aproximación.

La subdivisión de \mathcal{D} obtenida aplicando el algoritmo anterior con $\bar{E}(T) < \varepsilon$ será llamada ε - teselación afín por trozos de \mathcal{D} . Llamamos *precisión media (ap)* al valor $1/\varepsilon$.

5.4. Estructura del árbol neuronal basado en ASBC

En esta sección detallamos el paso 2 de los métodos indirectos para construir aproximantes neuronales en el caso del árbol de aproximación estudiado en las secciones 5.2 y 5.3. Este árbol está totalmente descrito por un vector estructura, cuyos componentes se llaman nodos. Hay dos clases de nodos: nodos hoja y no- hoja. Llamamos *nodo de computación* a la SLFN que aproxima uniformemente la función afín asociada a un nodo hoja y llamamos *nodo de direccionamiento* a la SLFN que representa la operación de un nodo no- hoja.

5.4.1. Nodos de cómputo

Dado $f : \mathcal{D} \rightarrow \mathbb{R}$, cada nodo hoja de una ε -PA teselación de \mathcal{D} tiene asociada a ella un d -simplex y una función afín $a_T(f)$ definida sobre él que difiere de f en ε (en valor promedio). Se ha probado en la sección 4.5 que una función afín definida sobre un d -simplex en \mathbb{R}^d puede ser aproximada uniformemente por una red neuronal de una capa con sólo dos neuronas y cuyos pesos se pueden expresar de forma analítica cerrada.

5.4.2. Nodos de direccionamiento

Cada valor de entrada \mathbf{x} debe ser direccionado al nodo de computación correspondiente. Esto puede ser conseguido reemplazando cada nodo no- hoja del árbol con una SLFN (nodo direccional) cuya estructura se describe más abajo.

ASBC es válido para dimensiones arbitrarias. Para simplificar, describiremos los nodos direccionales en el caso $d=3$. En otros casos el razonamiento es similar.

Supongamos que c es el tetraedro actual. Si este tetraedro es no-hoja, tiene dos hijos.

Sea $\mathbf{v} = N(N[c].l).\mathbf{v}$, entonces el tetraedro hijo izquierdo de c , se puede escribir como $T_l = \langle \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$. Cada cara de T_l se puede representar por la 3-combinación $\{i, j, k\}$ de $\{0, 1, 2, 3\}$. Denotemos por ζ el conjunto de 3-combinaciones de $\{0, 1, 2, 3\}$. La ecuación del plano que contiene la cara $\{i, j, k\}$ viene dada por

$$\mathbf{n}_{ijk} \cdot (\mathbf{x} - \mathbf{v}_i) = 0,$$

donde

$$\mathbf{n}_{ijk} \equiv (\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_i).$$

Definimos

$$E_{ijk} \equiv \mathbf{n}_{ijk} \cdot (\mathbf{v}_l - \mathbf{v}_i) \text{ where } l \notin \{i, j, k\} \text{ and } e_{ijk} \equiv E_{ijk} / |E_{ijk}|.$$

Los puntos \mathbf{x} dentro de T_l satisfacen la desigualdad

$$\mathbf{n}_{ijk} \cdot (\mathbf{x} - \mathbf{v}_i) e_{ijk} \geq 0 \text{ for all } \{ijk\} \in \zeta.$$

Consideremos la función de activación umbral

$$\varphi(t) = \begin{cases} 1, & t \geq 0, \\ 0, & t < 0. \end{cases}$$

Esta función se puede considerar como el límite de la función logística $s_p(t) = 1/(1 + e^{-pt})$ cuando el parámetro de la pendiente p tiende a infinito.

Definimos la SLFN direccional asociada al nodo c como

$$N_c^a(\mathbf{x}) \equiv \sum_{l=0}^3 \varphi(\mathbf{w}_l \cdot \mathbf{x} + b_l),$$

donde

$$\mathbf{w}_l = \mathbf{n}_{ijk} e_{ijk}, \quad (5.6)$$

$$b_l = -\mathbf{n}_{ijk} \cdot \mathbf{v}_i e_{ijk} \quad l \notin \{i, j, k\}. \quad (5.7)$$

5.4.3. Estructura del árbol resultante

Cada nodo del árbol tiene asociado un d -simplex y una SLFN perfectamente determinada. Si el nodo es una hoja, la SLFN es una red de computo, y una red de direccionamiento en caso contrario.

A partir de las secciones anteriores, podemos afirmar que el árbol neuronal correspondiente a una teselación que tiene n d -simplices consistirá en

$$(d+1)(n-1) + 2n = n(d+3) - d - 1$$

neuronas.

Llamamos $N_c^c(\mathbf{x})$ a la red de computo SLFN asociada con el nodo c (subsección 4.6). El algoritmo operativo de la red neuronal entrenada puede ser representado por el siguiente pseudocódigo tipo C.

```

Introduce los datos  $\mathbf{x}$ ,  $N[ ]$ .
 $c = 0$ ;
while (  $N[c].lfn == 0$ )
{
    if ( $N_c^a(\mathbf{x}) == 4$ )  $c = N[c].l$ ;
    else  $c = N[c].r$ ;
}
Calcula  $N_c^c(\mathbf{x})$ ;

```

De ahora en adelante, llamaremos árbol neuronal obtenido por división adaptativa basada en cubatura (NTASBC) al algoritmo descrito en las secciones 5.2, 5.3, 5.4.1 y 5.4.2.

La figura 5.2 muestra el árbol neuronal generado por NTASBC en el caso de la distancia al hexágono $cp6$ (ver sección 5.5) con $\varepsilon = 0.01$ y $ML = 0.001$ (748 triángulos). El dominio \mathcal{D} ha sido el triángulo de vértices $(-4,-4)$, $(4,-4)$ y $(4,4)$. Las redes SLFN de computo y direccionales no se dibujan explícitamente, pero están representadas por los nodos hoja y no-hoja del árbol, respectivamente.

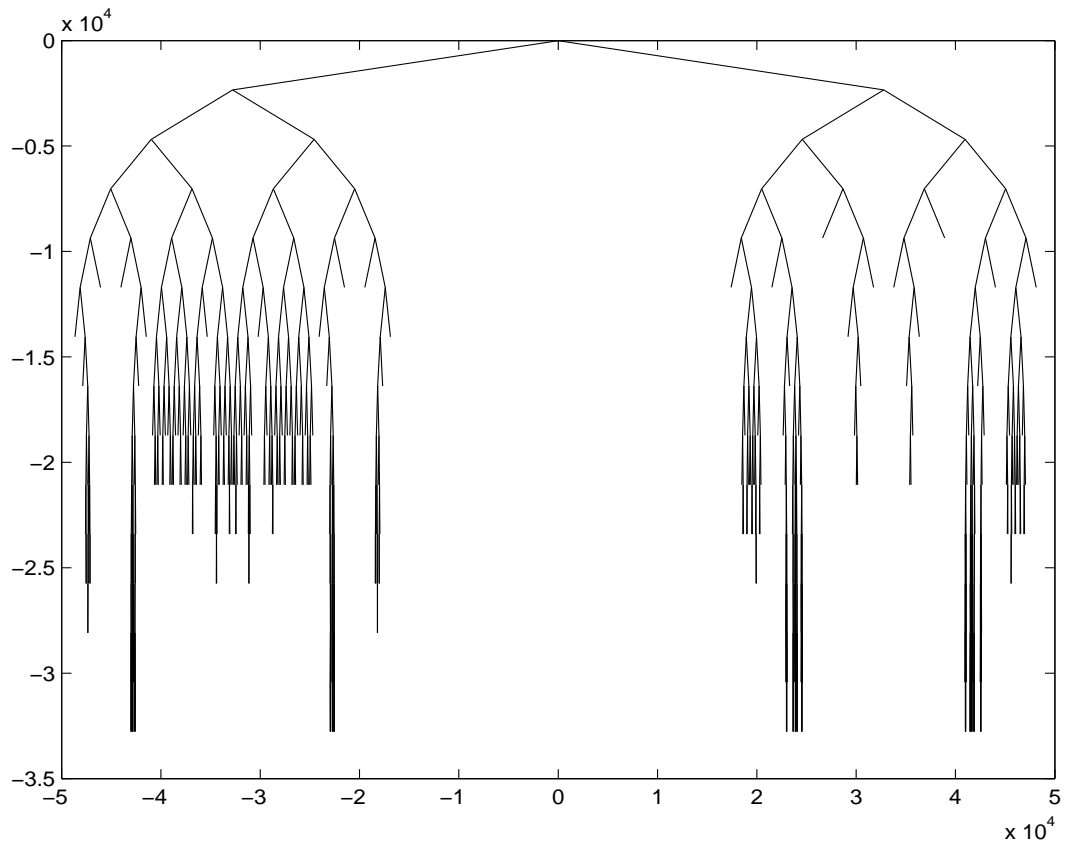


Figura 5.2: Árbol neuronal generado por NTASBC.

5.4.4. Computación biológica dirigida por datos

En las subsecciones anteriores hemos detallado los aspectos matemáticos y computacionales del árbol neuronal. Dado que estamos interesados en la plausibilidad biológica de los árboles neuronales ya descritos, en esta sección se propone un modelo para la computación dirigida por datos que tiene en cuenta algunos aspectos dinámicos del problema y la comunicación entre neuronas («spiking»). El modelo se basa en los siguientes conceptos y mecanismos:

- Redes neuronales «feedforward» de una capa.
- Tiempo de retardo de las señales en las neuronas.
- Neuronas inhibitorias.

Cuando la señal que codifica el valor de entrada x llega a una estructura direccional, debe realizar dos acciones:

1. Determinar el camino (izquierda o derecha) que la señal debe seguir.
2. Transmitir la señal inalterada a través de este camino.

El primer proceso usa la información contenida en la señal, pero en un contexto físico, esto puede modificar la señal. Una forma de resolver este problema consiste en enviar una señal duplicada x a cada estructura de direccionamiento (una estructura de direccionamiento consiste en una unidad de retardo (D) y una unidad inhibitoria (I)). La figura 5.3 muestra estas unidades en tres estructuras de direccionamiento del árbol.

Una copia de la señal se envía a la unidad de retardo. Esta unidad da, como salida, la señal de entrada inalterada con un pequeño tiempo de retardo. Durante este tiempo la otra copia de la señal llega a la unidad inhibitoria y da un valor negativo a los «biases» de las unidades de retardo derecha o izquierda correspondientes a las estructuras de direccionamiento hijas. Finalmente, las unidades de retardo transmiten la señal inalterada x a través del camino no

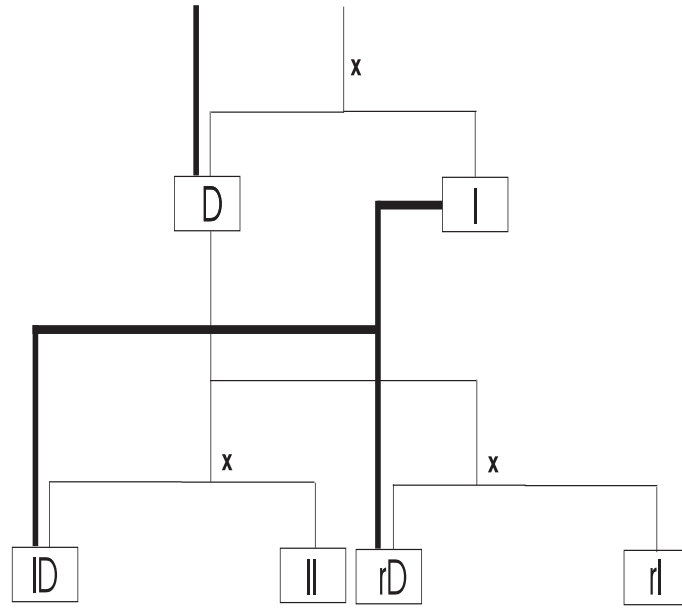


Figura 5.3: Direccionamiento biológico.

inhibido. A continuación describimos la estructura de las unidades de retardo e inhibitorias.

Una unidad de retardo está formada por tres SLFNs de retardo como se muestra en la figura 5.4. Las SLFNs de retardo están dadas por

$$N_i^D(x_i) = \sum_{j=1}^2 c_j^i s(W_j^i x_i + B_j^i), \quad i = 1, 2, 3.$$

Cada una de estas SLFNs producen una salida retardada idéntica a la entrada. Los pesos W_j^i y los «biases» B_j^i se determinan de acuerdo a la Proposición 1 de la sección 4.6.

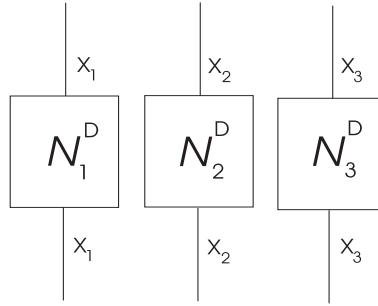


Figura 5.4: Unidad de retardo.

Las unidades inhibitorias consisten en una SLFN y en una neurona inhibitoria. La figura 5.5 muestra la estructura de este tipo de unidad.

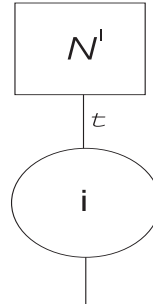


Figura 5.5: Unidad inhibitoria.

La SLFN inhibitoria está dada por

$$N^I(\mathbf{x}) = t = \sum_{j=0}^3 s_p(\mathbf{w}_j \cdot \mathbf{x} + b_j),$$

donde \mathbf{w}_j y b_j , $j = 0, 1, 2, 3$ están determinadas por las fórmulas (5.6) y (5.7). p es un número grande positivo.

La neurona inhibitoria muestra el siguiente comportamiento:

if $(t < 4 - \eta)$ $B_{lj}^i = NEG$, $j = 1, 2$, $i = 1, 2, 3$,

else $B_{rj}^i = NEG$, $j = 1, 2$, $i = 1, 2, 3$,

donde η es un número positivo pequeño, NEG es un número negativo con un valor absoluto grande y B_{lj}^i , B_{rj}^i son los «biases» de las unidades izquierda y derecha de retardo de las estructuras direccionales hijas de la estructura actual, respectivamente.

Los mecanismos anteriores tienen una base neurobiológica. Las células nerviosas individuales pueden multiplicar, integrar o retardar entradas sinápticas. Se ha observado la existencia de corrientes inhibitorias postsinápticas disparadas por la actividad (spiking) en la célula presináptica (Koch, 1999).

5.5. Comportamiento numérico de NTASBC.

En esta sección se ponen a prueba las capacidades de aprendizaje del algoritmo NTASBC para la función distancia euclídea a un cuerpo geométrico. Su cálculo rápido es necesario en aplicaciones robóticas (evitación de obstáculos, planificación de trayectorias, etc.) y en simulaciones de realidad virtual. En reconocimiento de patrones esta función permite calcular la distancia de Hausdorff entre politopos (Llanas, 2005) (La distancia de Hausdorff es una medida de similaridad entre cuerpos). Algunas aplicaciones de la representación neuronal de esta función han sido descritas, por ejemplo, en (Tseng, Hwang y Sheehan, 1997).

Para comparar las distancias obtenidas usando el árbol generado por NTASBC con las distancias exactas (calculadas usando el algoritmo LSABF (Llanas, Fernández de Sevilla y Feliú, 2000), hemos usado el error de generalización definido como la raíz cuadrada del error cuadrático medio (Nakashima y Ogawa, 2001).

$$RMSE \equiv \sqrt{\frac{\int_{\mathcal{D}} (d_a(\mathbf{x}) - d_e(\mathbf{x}))^2 d\mathbf{x}}{v(\mathcal{D})}}, \quad (5.8)$$

donde $d_a(\mathbf{x})$ y $d_e(\mathbf{x})$ son las distancias aproximadas y exactas desde el punto \mathbf{x} a la figura, respectivamente. \mathcal{D} denota el dominio de la función. Dado que el cálculo exacto de (5.8) es difícil, lo aproximamos usando un método de Monte Carlo. Este procedimiento elige una distribución de puntos uniformemente

aleatoria sobre \mathcal{D} y calcula (5.8) como

$$RMSE \simeq \sqrt{\frac{\sum_{j=1}^N (d_a(\mathbf{x}_j) - d_e(\mathbf{x}_j))^2}{N}}. \quad (5.9)$$

En la práctica, hemos promediado la parte derecha de (5.9) sobre 50 ensayos. Cada ensayo consiste en una distribución uniforme al azar de N puntos sobre \mathcal{D} . El valor resultante se denotará por RMSE de ahora en adelante. También incluimos la desviación estándar de este valor.

5.5.1. Experimentos en dos dimensiones

Hemos aplicado NTASBC al aprendizaje de la función distancia desde un punto variable a una figura en el plano. El dominio de esta función ha sido $\mathcal{D} = [-4, 4]^2$, esto es, un cuadrado que rodea cada figura. Este cuadrado se divide en dos triángulos usando la diagonal principal. NTASBC ha sido aplicado sucesivamente a cada triángulo. Los modelos bidimensionales se muestran en la figura 5.6. Usamos la siguiente notación:

- cp6: hexágono convexo.
- cp20: polígono convexo de 20 lados.
- c: círculo de radio 2.
- ncp: polígono no convexo de 18 lados.

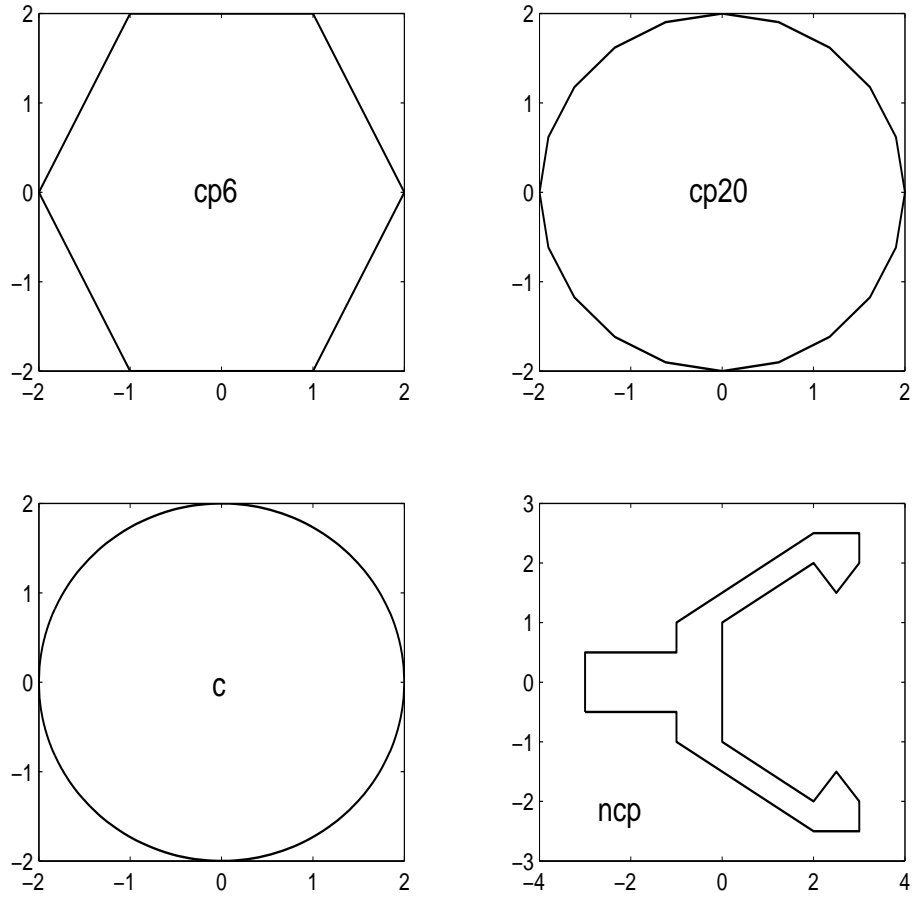


Figura 5.6: Figuras usadas en los experimentos 2D.

Los resultados en las Tablas 5.1, 5.2, y 5.3 se muestran en la Figura 5.7 como gráficos de precisión media frente al número de triángulos en la teselación, RMSE ($N = 10000$) y el tiempo de entrenamiento. Hacemos notar el decrecimiento hiperbólico de RMSE cuando la precisión media se incrementa. Un hecho notable ha sido la gran velocidad de entrenamiento de NTASBC (el tiempo de entrenamiento ha sido menor de 50 segundos en todos los casos considerados).

figura 2D	<i>ap</i>			
	10	100	1000	10000
cp6	60	748	6772	60316
cp20	80	984	10600	104712
c	72	872	9472	100880
ncp	164	1500	15204	149200

Tabla 5.1: Número de triángulos en las teselaciones 2D.

figura 2D	<i>ap</i>			
	10	100	1000	10000
cp6	0.02	0.19	1.62	13.95
cp20	0.06	0.44	4.67	40.76
c	0.01	0.09	0.83	8.81
ncp	0.13	1.07	9.91	95.17

Tabla 5.2: Tiempo de CPU para el entrenamiento (en segundos).

2D	<i>ap</i>							
	10		100		1000		10000	
	RMSE	Dev	RMSE	Dev	RMSE	Dev	RMSE	Dev
cp6	0.075329	$0.604 \cdot 10^{-3}$	0.005334	$0.612 \cdot 10^{-4}$	0.000428	$0.617 \cdot 10^{-5}$	0.000039	$0.430 \cdot 10^{-6}$
cp20	0.058186	$0.439 \cdot 10^{-3}$	0.005747	$0.393 \cdot 10^{-4}$	0.000475	$0.483 \cdot 10^{-5}$	0.000047	$0.477 \cdot 10^{-6}$
c	0.057412	$0.458 \cdot 10^{-3}$	0.006607	$0.368 \cdot 10^{-4}$	0.000633	$0.359 \cdot 10^{-5}$	0.000060	$0.427 \cdot 10^{-6}$
ncp	0.063008	$0.696 \cdot 10^{-3}$	0.004355	$0.610 \cdot 10^{-4}$	0.000381	$0.562 \cdot 10^{-5}$	0.000040	$0.510 \cdot 10^{-6}$

Tabla 5.3: Error de generalización en 2D de los árboles generados por NTASBC.

La figura 5.8 muestra los gráficos de las teselaciones obtenidas aplicando NTASBC a la función distancia al hexágono cp6 y al polígono no convexo ncp. El número de triángulos representados en los gráficos es

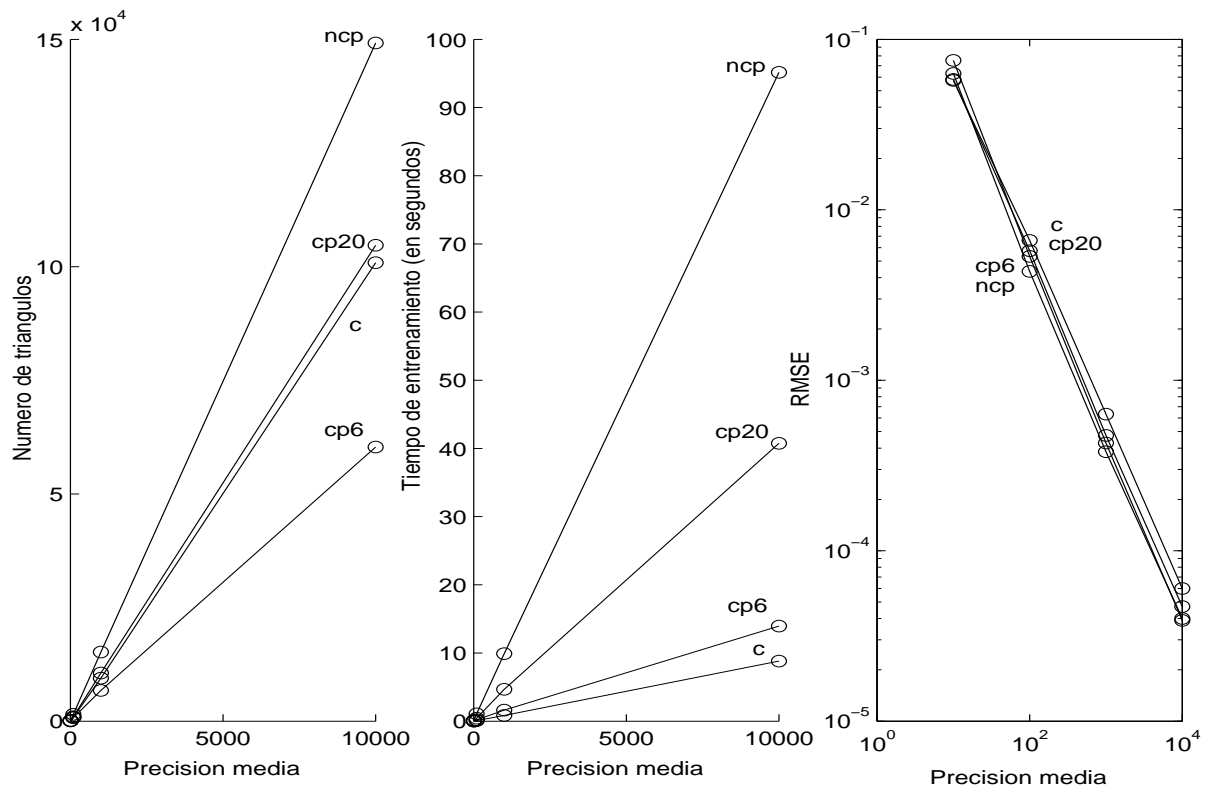


Figura 5.7: Resultados de NTASBC en dos dimensiones.

- hexágono con precisión media 100: 748 triángulos.
- hexágono con precisión media 1000: 6772 triángulos.
- polígono no convexo con precisión media 100: 1500 triángulos.
- polígono no convexo con precisión media 1000: 15204 triángulos.

Estos gráficos muestran que la densidad de triángulos de la teselación se incrementa en la vecindad de la frontera de cada figura considerada. Esto es debido al hecho de que el gradiente de función distancia es discontinuo en ese lugar. Esto

hace difícil la aproximación en ambos lados de la frontera por una única función afín.

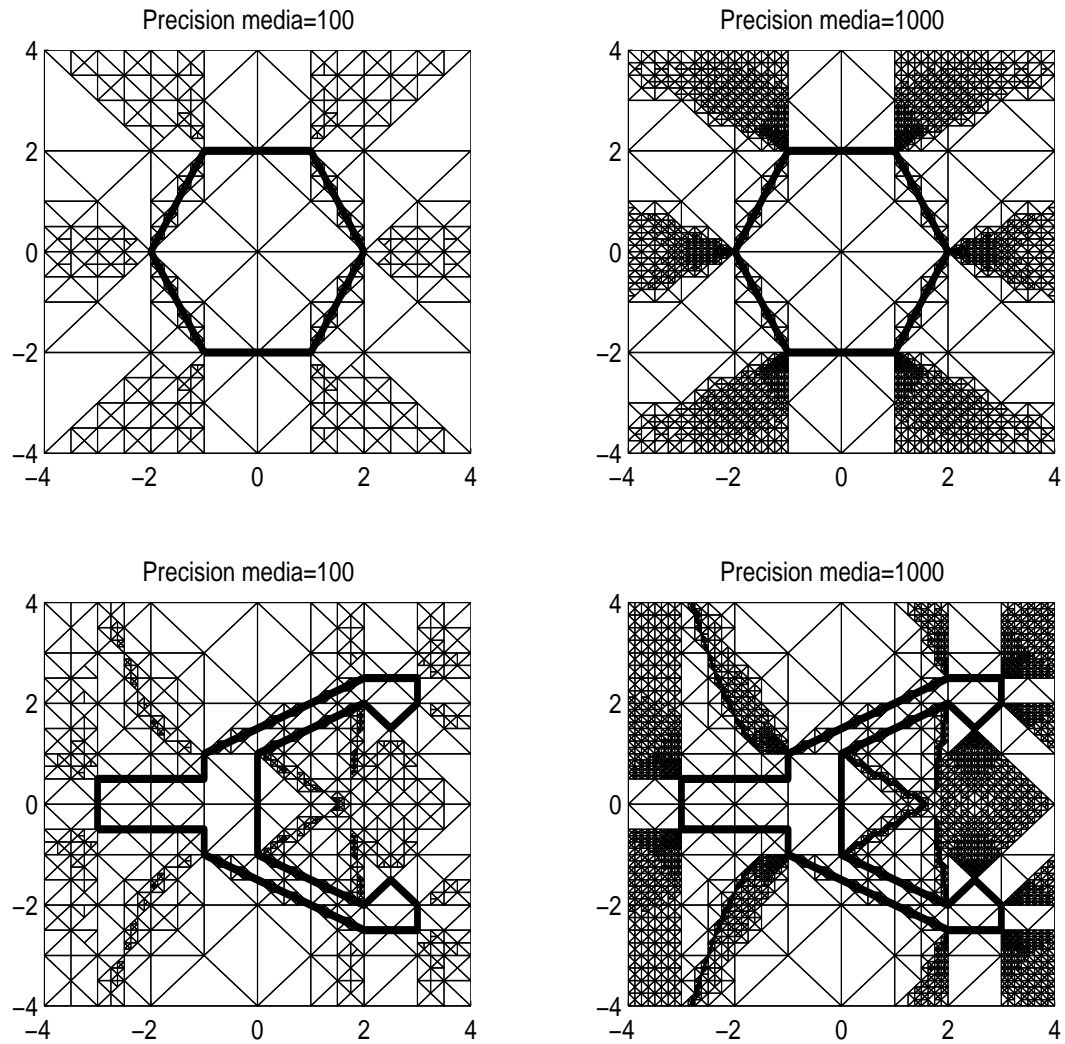


Figura 5.8: Teselaciones generadas por NTASBC en el caso del hexágono y del polígono no convexo.

5.5.2. Experimentos en tres dimensiones

Se ha aplicado NTASBC al aprendizaje de la función distancia desde un punto variable a cuerpos en tres dimensiones. El dominio ha sido el cubo $\mathcal{D} = [-4, 4]^3$ que contiene cada cuerpo. Hemos dividido cada una de las seis caras del cubo en dos triángulos, utilizando sus diagonales. Usando esta división, el cubo se ha descompuesto en doce tetraedros congruentes cuyo vértice común es el centro del cubo. NTASBC ha sido aplicado sucesivamente a cada tetraedro. En el caso del poliedro no convexo, \mathcal{D} ha sido dividido en 30 tetraedros debido a limitaciones de la memoria RAM. En algunos experimentos 3D hemos usado poliedros elipsoidales, esto es, aquellos inscritos en el elipsoide

$$x^2/a^2 + y^2/b^2 + (z - c)^2/c^2 = 1.$$

El elipsoide ha sido subdividido axialmente (eje z) en m partes y radialmente (plano x - y) en n partes. El poliedro resultante tiene $(m - 2)n + 2$ vértices y $(m - 1)n$ caras. Denotamos por $\mathcal{E}(n, m, a, b, c)$ cualquier poliedro de este tipo. Los modelos 3D se muestran en las Figuras 5.9 y 5.10. Hemos usado la siguiente notación:

- cpl4: tetraedro.
- cpl6: cubo.
- cpl20: icosaedro.
- epl1: $\mathcal{E}(10, 10, 1, 1, 1.5)$ (82 vértices y 90 caras).
- epl2: $\mathcal{E}(20, 26, 1, 1, 1.5)$ (482 vértices y 500 caras).

- epl3: $\mathcal{E}(25, 33, 1, 1, 1.5)$ (777 vértices y 800 caras).
- epl4: $\mathcal{E}(32, 34, 1, 1, 1.5)$ (1026 vértices y 1056 caras).
- b: esfera de radio 1.5 centrada en el origen.
- ncpl: poliedro no convexo.

Los datos específicos de algunos de estos poliedros se detallan seguidamente.

- Los vértices del tetraedro son:

(1,5,-1,5,-1,5)	(1,5,1,5,-1,5)	(-1,5,0,-1,5)	(0,0,1,5)
-----------------	----------------	---------------	-----------

- Los vértices del cubo son:

(1.5,-1.5,-1.5)	(1.5,1.5,-1.5)	(-1.5,1.5,-1.5)	(-1.5,-1.5,-1.5)
(1.5,-1.5,1.5)	(1.5,1.5,1.5)	(-1.5,1.5,1.5)	(-1.5,-1.5,1.5)

- Los vértices del icosaedro son:

(1,0, τ)	(-1,0, τ)	(0,- τ ,1)	(τ ,-1,0)	(τ ,1,0)	(0, τ ,1)
(- τ ,-1,0)	(- τ ,1,0)	(0, τ ,-1)	(-1,0,- τ)	(1,0,- τ)	(0,- τ ,-1)

donde $\tau=1.618034$.

- El poliedro no convexo es la unión de tres paralelepípedos cuyos vértices son:

$(1.5, -1.5, -1.5)$	$(1.5, -1., -1.5)$	$(-0.5, -1., -1.5)$	$(-0.5, -1.5, -1.5)$
$(1.5, -1.5, -0.5)$	$(1.5, -1., -0.5)$	$(-0.5, -1., -0.5)$	$(-0.5, -1.5, -0.5)$

$(0., -1., -1.5)$	$(0., 1.5, -1.5)$	$(-0.5, 1.5, -1.5)$	$(-0.5, -1., -1.5)$
$(0., -1., -0.5)$	$(0., 1.5, -0.5)$	$(-0.5, 1.5, -0.5)$	$(-0.5, -1., -0.5)$

$(0., 1., -0.5)$	$(0., 1.5, -0.5)$	$(-0.5, 1.5, -0.5)$	$(-0.5, 1., -0.5)$
$(0., 1., 1.5)$	$(0., 1.5, 1.5)$	$(-0.5, 1.5, 1.5)$	$(-0.5, 1., 1.5)$

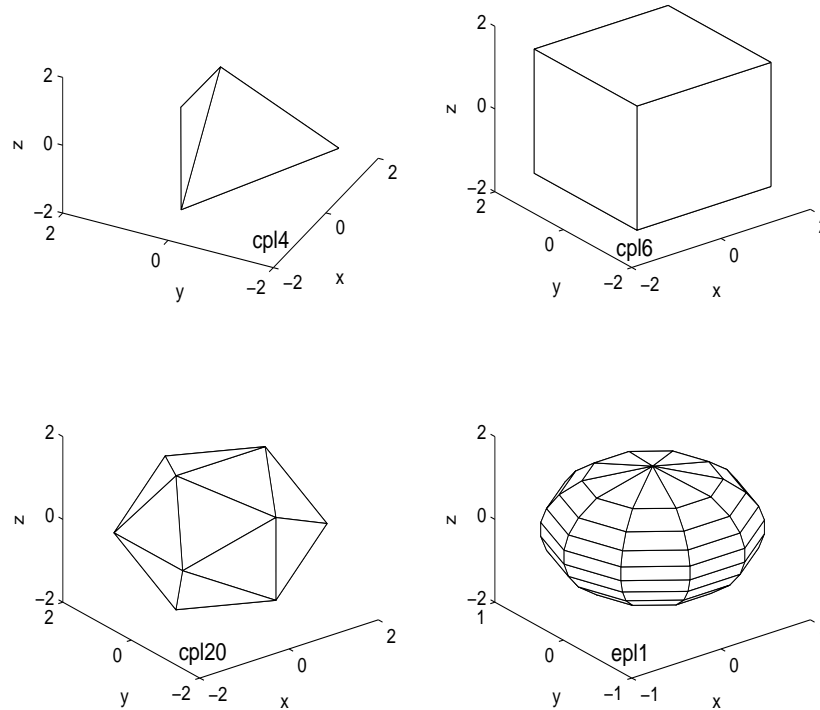


Figura 5.9: Poliedros convexos utilizados en los experimentos 3D.

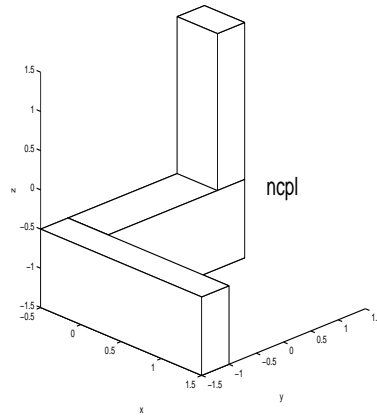


Figura 5.10: Poliedro no convexo utilizado en los experimentos 3D.

cuerpo 3D	ap			
	10	100	200	1000
cpl4	1892	69289	216928	3467274
cpl6	2112	60384	169680	2001696
cpl20	2184	89976	296424	6283080
epl1	1808	72768	219424	3562930
epl2	1760	69680	217872	3579664
epl3	1752	69138	214577	3551099
epl4	1744	65888	207252	3522608
b	1776	77328	251904	4468128
ncpl	2276	108093	379256	7840660

Tabla 5.4: Número de tetraedros en las teselaciones 3D.

cuerpo 3D	<i>ap</i>			
	10	100	200	1000
cpl4	1.55	53.39	167.30	2828.28
cpl6	2.97	64.25	180.12	2065.13
cpl20	5.61	192.69	599.46	13455.74
epl1	8.66	281.36	817.18	12640.18
epl2	13.59	586.26	1945.31	34731.72
epl3	15.87	783.12	2618.11	48946.90
epl4	19.10	935.61	3496.08	60492.77
b	0.13	6.57	23.42	473.91
ncpl	14.30	628.18	2384.55	82459.76

Tabla 5.5: Tiempo de CPU para el entrenamiento (en segundos).

3D	<i>ap</i>							
	10		100		200		1000	
	RMSE	Dev	RMSE	Dev	RMSE	Dev	RMSE	Dev
cpl4	0.077025	$0.834 \cdot 10^{-4}$	0.007448	$0.857 \cdot 10^{-5}$	0.003765	$0.412 \cdot 10^{-5}$	0.000744	$0.799 \cdot 10^{-6}$
cpl6	0.081633	$0.899 \cdot 10^{-4}$	0.006425	$0.106 \cdot 10^{-5}$	0.003284	$0.413 \cdot 10^{-5}$	0.000649	$0.987 \cdot 10^{-6}$
cpl20	0.068098	$0.673 \cdot 10^{-4}$	0.007177	$0.827 \cdot 10^{-5}$	0.003730	$0.453 \cdot 10^{-5}$	0.000720	$0.113 \cdot 10^{-5}$
epl1	0.078830	$0.822 \cdot 10^{-4}$	0.007889	$0.649 \cdot 10^{-5}$	0.003936	$0.372 \cdot 10^{-5}$	0.000784	$0.713 \cdot 10^{-6}$
epl2	0.075616	$0.726 \cdot 10^{-4}$	0.007866	$0.575 \cdot 10^{-5}$	0.004000	$0.373 \cdot 10^{-5}$	0.000791	$0.750 \cdot 10^{-6}$
epl3	0.075991	$0.760 \cdot 10^{-4}$	0.007923	$0.743 \cdot 10^{-5}$	0.003998	$0.362 \cdot 10^{-5}$	0.000796	$0.752 \cdot 10^{-6}$
epl4	0.075894	$0.745 \cdot 10^{-4}$	0.008080	$0.636 \cdot 10^{-5}$	0.004031	$0.328 \cdot 10^{-5}$	0.000799	$0.751 \cdot 10^{-6}$
b	0.071815	$0.690 \cdot 10^{-4}$	0.007795	$0.801 \cdot 10^{-5}$	0.003971	$0.312 \cdot 10^{-5}$	0.000818	$0.780 \cdot 10^{-6}$
ncpl	0.085477	$0.101 \cdot 10^{-3}$	0.007538	$0.953 \cdot 10^{-5}$	0.003801	$0.446 \cdot 10^{-5}$	0.000757	$0.954 \cdot 10^{-6}$

Tabla 5.6: Error de generalización en 3D de los árboles generados por NTASBC.

Los resultados en las Tablas 5.4, 5.5 y 5.6 se muestran en las Figuras 5.11 y 5.12 como gráficos de precisión media frente al número de tetraedros en la teselación y el tiempo de entrenamiento, y precisión promedio frente a RMSE ($N = 200000$), respectivamente. Como en el caso 2D, hay que notar el decrecimiento de RMSE de acuerdo a la ley $RMSE = \kappa / ap$, donde κ es una constante menor que 1. Los experimentos sobre poliedros elípticos con precisión media fijada muestran teselaciones que tienen un número casi constante de tetraedros independientemente de la complejidad combinatoria del poliedro considerado.

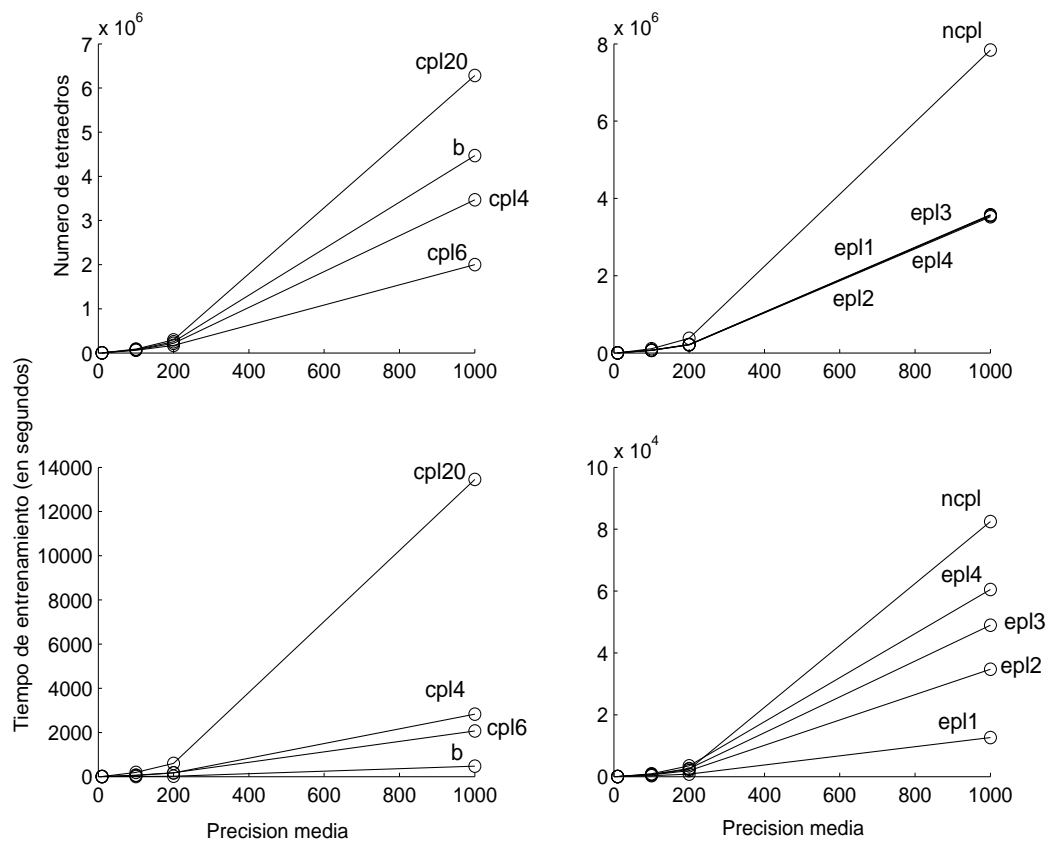


Figura 5.11: Resultados de NTASBC en tres dimensiones.

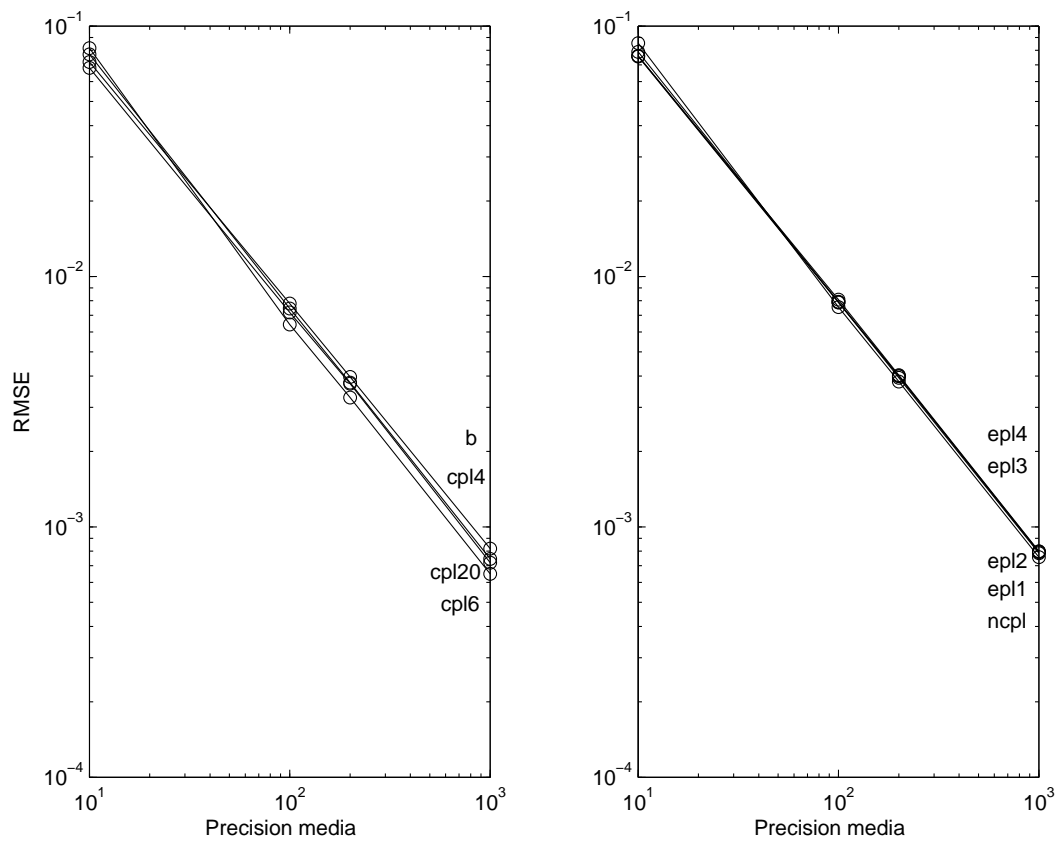


Figura 5.12: Error de generalización de los árboles generados por NTASBC en tres dimensiones.

La figura 5.13 muestra los gráficos que resultan de la intersección de planos horizontales con las teselaciones generadas por NTASBC correspondientes al icosaedro cpl20.

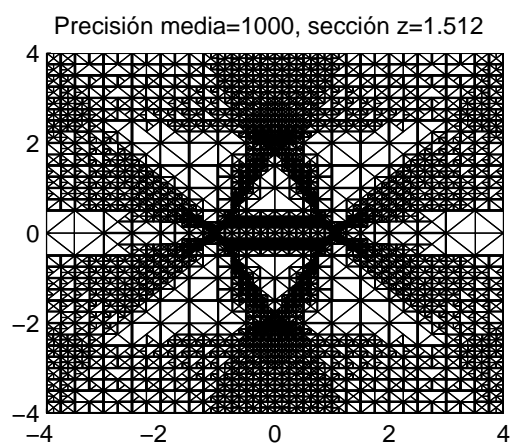
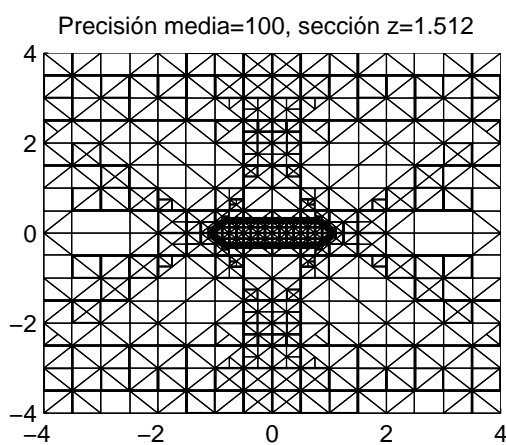
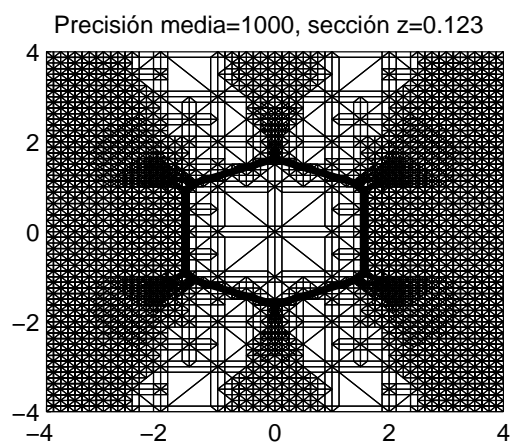
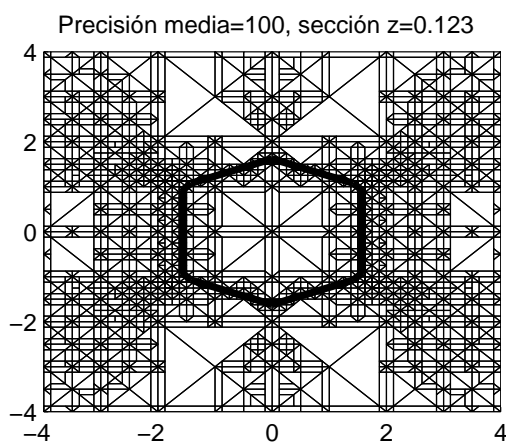


Figura 5.13: Intersección de planos horizontales con las teselaciones generadas por NTASBC.

5.5.3. Número de distancias calculadas por NTASBC

En la Tabla 5.7 detallamos el número de distancias calculadas por NTASBC en el entrenamiento correspondiente a varios objetos tridimensionales.

ap	epl1	epl2	epl3	epl4	b
10	285128	266328	261172	272616	248916
100	11807764	11739256	12095914	11790312	13654068
200	36761212	39601108	39653368	38904528	49311444
1000	699242462	727003740	728189286	724676944	1018961412

Tabla 5.7: Número de distancias calculadas por NTASBC.

5.5.4. Experimentos de variación de dominio

En la mayor parte de las aplicaciones: coger objetos, evitación de obstáculos, etc., es suficiente construir la función distancia en un dominio de tamaño moderado que rodee al cuerpo. Hemos estudiado el incremento en el número de tetraedros en las teselaciones cuando el dominio de la función distancia crece.

Los resultados obtenidos al aplicar NTASBC a las distancias a cpl4, cpl20 y epl3 se muestran como gráficos de volumen frente a número de tetraedros en la teselación en la figura 5.14. En todos los casos, los parámetros de los algoritmos han sido $\varepsilon = 10^{-2}$ y $ML = 10^{-3}$. Se han considerado los dominios $\mathcal{D} = [-3, 3]^3$, $[-4, 4]^3$, $[-5, 5]^3$ y $[-6, 6]^3$.

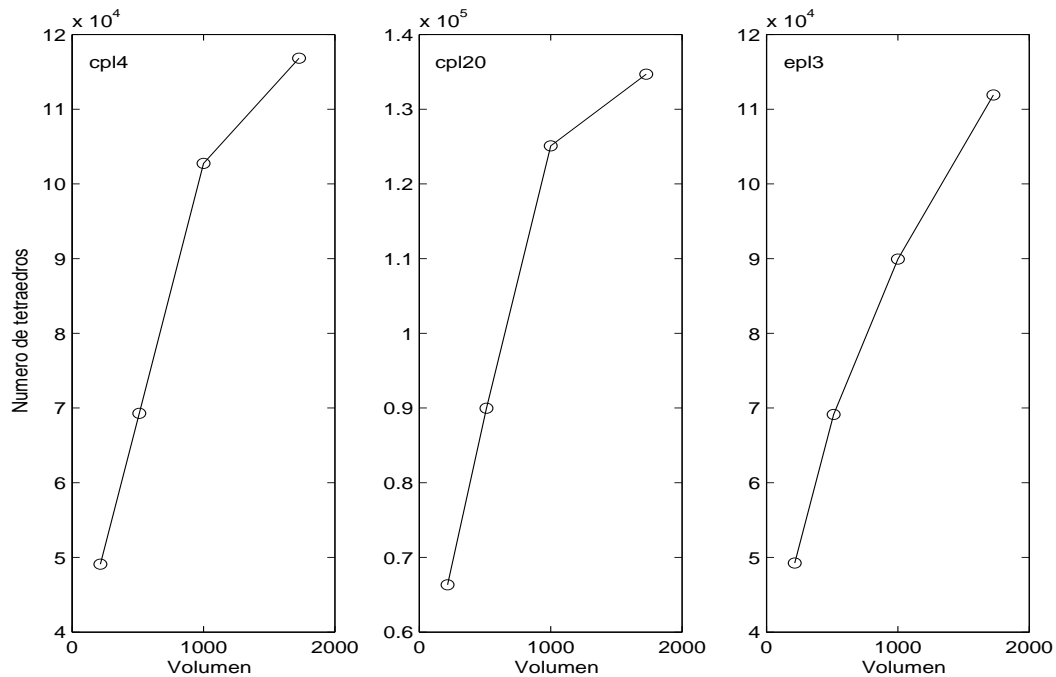


Figura 5.14: Experimentos de variación de dominio.

5.6. Comparación entre redes obtenidas mediante NTASBC y redes SLFN

5.6.1. Comparación entre redes obtenidas mediante NTASBC y redes SLFN entrenadas por un método de regiones de confianza basado en el algoritmo interior-reflexivo de Newton

En esta sección comparamos los mejores resultados de generalización obtenidos usando el árbol generado por NTASBC con aquellos obtenidos por

redes SLFN logísticas entrenadas por la función *lsqnonlin* de la «toolbox» de optimización de MATLAB. Por defecto *lsqnonlin* elige el algoritmo para problemas de alta complejidad. Este algoritmo (STRIRN) es un método de región de confianza y está basado en el algoritmo de Newton interior-reflexivo descrito en (Coleman y Li, 1994) y (Coleman y Li, 1996).

En el caso de figuras 2D hemos entrenado diferentes SLFNs sobre un conjunto de datos correspondientes a una malla regular rectangular sobre $\mathcal{D} = [-4, 4]^2$ con $60 \times 60 = 3600$ puntos. El número de neuronas en la capa oculta ha sido: 10, 15, 20 y 25. Los mejores resultados corresponden a 25 neuronas en la capa oculta. En el caso de cuerpos tridimensionales, hemos entrenado diferentes SLFNs con el conjunto de datos correspondiente a una malla rectangular sobre $\mathcal{D} = [-4, 4]^3$ con $20 \times 20 \times 20 = 8000$ puntos. Hemos considerado 10, 15 y 25 neuronas en la capa oculta en todos los casos. Los mejores resultados corresponden a 25 neuronas en la capa oculta. En ambos casos, 2D y 3D, hemos fijado un número máximo de evaluaciones de la función objetivo: 30000. El error cuadrático medio se ha calculado usando el mismo procedimiento y los conjuntos de prueba descritos en la sección previa. Los mejores resultados se dan en las Tablas 5.8 y 5.9.

Los resultados en las tablas 5.8 y 5.9 se muestran en las figuras 5.15 y 5.16 como gráficos de tipo de figura frente al cociente del mejor RSME de NTASBC por el correspondiente a la red SLFN entrenada, y tipo de figura frente al cociente del mejor tiempo de entrenamiento de NTASBC por el correspondiente a la SLFN entrenada.

figura 2D	Entrenamiento	Cálculo		Tiempo (s)
	RMSE	RMSE	Dev	Entrenamiento
cp6	0.029655	0.029782	0.000331	95783.6
cp20	0.021339	0.021172	0.000202	95759.9
circle	0.023327	0.023231	0.000240	98842.7
ncp	0.038145	0.038116	0.000271	95414.8

Tabla 5.8: Resultados de aprendizaje de redes SLFN entrenadas por STRIRN en el caso bidimensional.

cuerpo 3D	Entrenamiento	Cálculo		Tiempo (s)
	RMSE	RMSE	Dev	Entrenamiento
cpl4	0.040800	0.039997	0.000117	213618.4
cpl6	0.068366	0.068469	0.000186	276574.3
cpl20	0.062063	0.064823	0.000207	251478.7
epl1	0.035809	0.023097	0.000104	165657.3
epl2	0.019769	0.020416	0.000126	257396.4
epl3	0.020840	0.021271	0.000126	258116.2
epl4	0.019998	0.020436	0.000137	255706.0
b	0.030773	0.032334	0.000196	260263.8
ncpl	0.065805	0.066339	0.000178	260281.5

Tabla 5.9: Resultados de aprendizaje de redes SLFN entrenadas por STRIRN en el caso tridimensional.

El tiempo de CPU para el entrenamiento de NTASBC es significativamente más pequeño que el correspondiente a STRIRN, incluso considerando que el error de generalización de los árboles generados por NTASBC es más de veinticinco veces menor que el obtenido por las redes SLFN. El mejor error de generalización

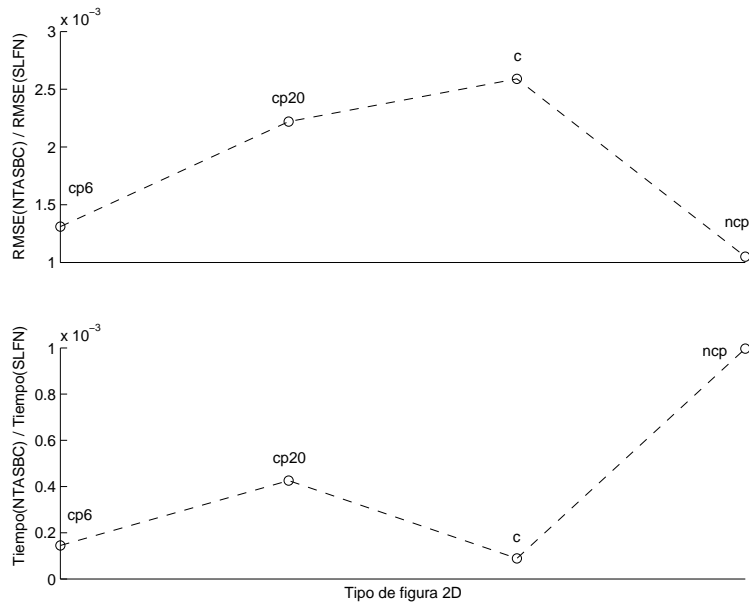


Figura 5.15: Comparación entre NTASBC y SLFNs entrenadas con STRIRN en el caso de figuras bidimensionales.

obtenido por redes SLFN entrenadas con STRIRN puede ser obtenido con árboles generados por NTASBC con un tiempo de entrenamiento entre 300 y 100000 veces más bajo (Tablas 5.2, 5.3, 5.8, 5.5, 5.6 y 5.9).

El error de generalización de una red SLFN entrenada por un procedimiento de interpolación puede ser reducido incrementando el número de datos de entrenamiento y de neuronas en la capa oculta. El incremento en el número de datos de entrenamiento hace que la función de error sea más compleja. El incremento en el número de neuronas incrementa el número de variables de la función de error. Por tanto, cuando entrenamos una red SLFN usando un método de optimización iterativo, los incrementos anteriores implican un crecimiento considerable del tiempo de entrenamiento. Hemos hecho experimentos con el

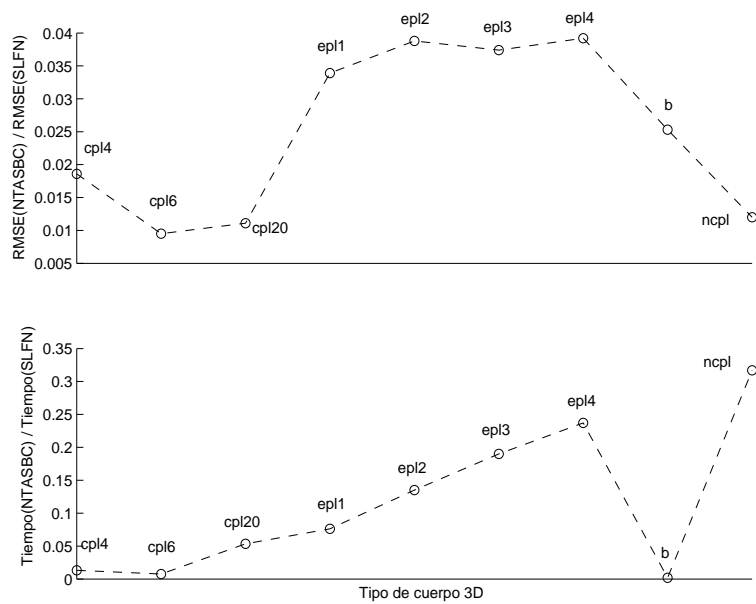


Figura 5.16: Comparación entre NTASBC y SLFNs entrenadas con STRIRN en el caso de figuras tridimensionales.

poliedro ep14 cuyos resultados pueden verse en las tablas 5.10, 5.11 y en la figura 5.17.

Número de datos de entrenamiento	Tiempo de entrenamiento(s)
1000	030210.95
4096	128233.55
8000	255706.00

Tabla 5.10: Tiempo de entrenamiento con un número fijo de neuronas (nn=25).

Número de neuronas ocultas	Tiempo de entrenamiento(s)
10	05119.7
15	19018.1
25	30210.9

Tabla 5.11: Tiempo de entrenamiento con un número fijo de datos de entrenamiento (ntd=1000).

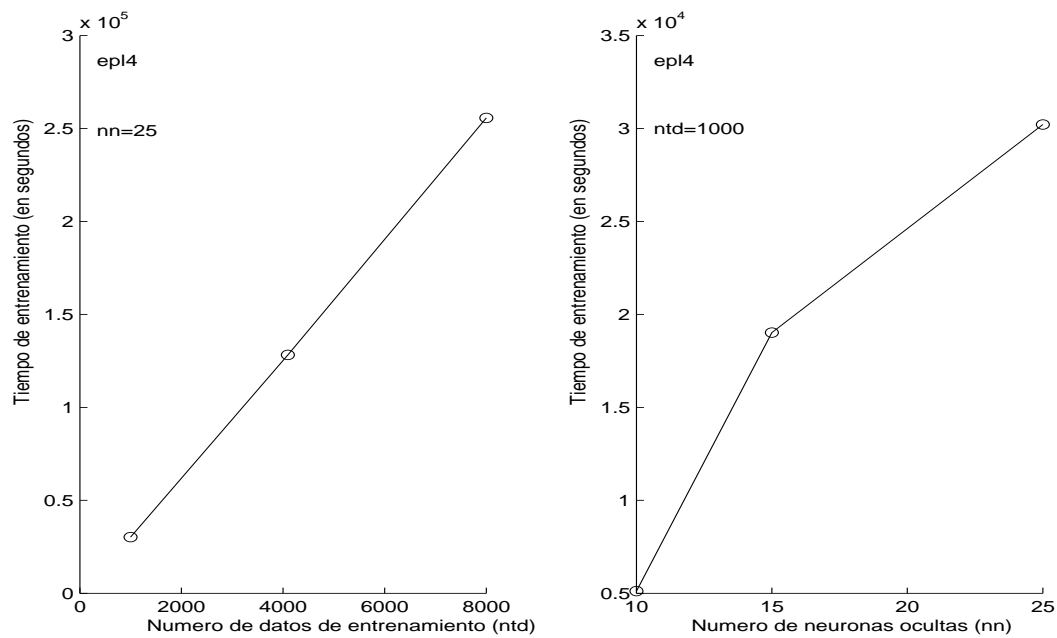


Figura 5.17: Crecimiento del tiempo de entrenamiento de SLFNs entrenadas con STRIRN.

En contraste, NTASBC puede integrar una enorme cantidad de datos de entrenamiento (ver subsección 5.5.3) de una forma rápida y fácil. Consecuentemente, el producto de su error de generalización por el tiempo de entrenamiento es inferior al correspondiente a redes del tipo SLFN entrenadas mediante STRIRN.

5.6.2. Tiempos de cálculo individuales

En esta sección mostramos el tiempo de cálculo de la distancia de un punto a un cuerpo 3D evaluado con los algoritmos y redes anteriormente descritos. El tiempo de cálculo ha sido promediado sobre ocho puntos escogidos aleatoriamente. Los resultados se dan en las tablas 5.12 y 5.13.

Algoritmo	epl1	epl2	epl3	epl4	b
Cálculo directo					0.27
LSABF	31.40	54.85	58.83	80.61	
SLFN (25 neuronas)	9.99	9.97	9.95	10.05	10.03

Tabla 5.12: Tiempo medio de cálculo de distancias individuales (en microsegundos).

ap	epl1	epl2	epl3	epl4	b
10	14.92	15.54	14.93	14.99	15.82
100	22.45	23.17	22.75	22.31	21.89
200	24.68	24.33	24.49	24.51	23.58
1000	27.83	29.48	29.35	29.95	25.98

Tabla 5.13: Tiempo medio de cálculo de distancias individuales (en microsegundos) de los árboles generados por NTASBC.

El tiempo de cálculo de distancias individuales con los árboles generados por NTASBC es inferior al correspondiente a uno de los métodos exactos más rápidos (LSABF). Las redes SLFN con 25 neuronas ocultas, tienen un tiempo de cálculo

inferior que los árboles neuronales generados por NTASBC (Tablas 5.12 y 5.13), pero su error de generalización es mucho más grande.

De los resultados anteriores podemos concluir que la mayor parte del tiempo de CPU para el entrenamiento de NTASBC se dedica a la computación de la función considerada. Por ejemplo, en el caso de la esfera, tomando $ap = 1000$, y usando los resultados en las Tablas 5.5, 5.7, y 5.12, tenemos

- Tiempo de entrenamiento total: 473.91 s.
- Tiempo dedicado a computar distancias: 275.12 s.

Es decir, 58.05 % del tiempo de CPU está dedicado a la computación de las distancias. En el caso de animales y robots, los datos se obtienen por órganos sensoriales o sensores que pueden muestrear el espacio de entrada de forma paralela. Esto acelera el algoritmo pues disminuye el tiempo de muestreo de la de la función a aprender.

5.7. Conclusiones

En muchos problemas de aprendizaje natural, el aprendiz tiene la capacidad de actuar sobre su entorno y conseguir datos para optimizar su adquisición de conocimiento. Estos datos deben ser seleccionados cuidadosamente para construir un buen modelo.

Una gran parte de la investigación sobre aprendizaje en hombres y máquinas, trata al sujeto que aprende como un recipiente pasivo de información e ignora el papel del componente activo del aprendizaje. En esta tesis se muestra un método activo de aprendizaje para construir árboles neuronales.

Un hecho notable es que una red SLFN con sólo dos neuronas puede aproximar uniformemente una función afín sobre un d -simplex, independientemente de d . Esto nos permite diseñar una red neuronal estructurada en forma de árbol y autogenerada basada en el algoritmo de NTASBC. El algoritmo NTASBC consta de dos etapas:

- Una aproximación afín por trozos obtenida por división adaptativa basada en cubatura (ASBC).
- El reemplazamiento de los nodos no-hoja y hoja del árbol de aproximación con redes SLFN de direccionamiento y de computación, respectivamente.

El algoritmo NTASBC tiene varias características deseables:

- Muestreo automático del espacio de entrada para obtener buenos resultados de aprendizaje. La técnica de cubatura permite integrar una gran cantidad de información acerca de la función objetivo de una forma rápida.
- La topología y los pesos del árbol neuronal son generados por un método de división adaptativa y el procedimiento expuesto en la Proposición 1 y el Teorema 4. No es necesario un entrenamiento adicional.

- Los errores de entrada tienen sólo un efecto local sobre el árbol resultante, es decir, la respuesta del árbol permanece inalterada en las subregiones adecuadamente muestreadas del dominio.

Estas características son adecuadas para el aprendizaje a través de un sistema que puede muestrear el espacio circundante mediante órganos sensoriales. En los ejemplos numéricos hemos considerado el aprendizaje de la función distancia a objetos bidimensionales y tridimensionales. Esta función proporciona una representación del espacio que rodea a los objetos físicos y facilita movimientos sin colisión. Este tipo de representación podría ser efectuada dentro del córtex parietal.

El algoritmo propuesto no está basado en un método de interpolación. Requiere un conocimiento completo de la función objetivo, es decir, debemos ser capaces de disponer del valor de la función en cualquier punto del dominio considerado.

Aunque en la práctica con redes neuronales son habituales funciones que sólo se conocen en algunos puntos de su dominio, también aparecen en las aplicaciones funciones que están completamente definidas. Por ejemplo, si disponemos de:

1. Una expresión analítica de f .
2. Un algoritmo para computar f .
3. Un dispositivo físico que suministra los valores de f (sensores,...).

La aproximación de funciones completamente conocidas por redes neuronales tiene un interés práctico, porque puede suministrar:

- Aproximantes analíticos a f en los casos 2 y 3.
- Una forma más rápida de evaluar f (aplicaciones en tiempo real).
- Modelos teóricos de aprendizaje biológico.

Los experimentos computacionales muestran la eficiencia de aprendizaje del algoritmo propuesto:

- Para un error de generalización fijo, NTASBC es entre 300 y 100000 veces más rápido que el algoritmo de región de confianza de Newton interior-reflexivo aplicado a redes SLFN. El tiempo de entrenamiento de NTASBC se podría reducir usando dispositivos más rápidos de adquisición de datos.
- El error de generalización puede ser fijado a priori usando el concepto de precisión media. Ambas variables están relacionadas por una ley hiperbólica con una constante menor que 1 (sección 5.5).
- Los tiempos medios de cálculo de distancias individuales de los árboles generados por NTASBC son inferiores a aquellos utilizados por el algoritmo LSABF.
- Los resultados de la subsección 5.5.3 sugieren que el número de neuronas de la representación del árbol neuronal de la distancia a un poliedro convexo, no depende de su complejidad combinatoria.

En resumen, NTASBC proporciona un procedimiento simple y rápido de construcción de árboles neuronales para aprender con precisión la función distancia desde un punto variable a un cuerpo geométrico. Esta capacidad parece extensible a funciones moderadamente no lineales.

Conclusión

El propósito principal de este trabajo es construir un modelo de redes neuronales de la representación cognitiva de un objeto en el espacio para actuar sobre él, como en el caso de cogerlo adecuadamente. Previamente a este objetivo hemos hecho un recorrido por el aprendizaje biológico de neuronas reales, por la base neurobiológica de la acción en el córtex y por la definición e historia de las MLFNs.

El proceso de construcción de campos de distancia a objetos en el espacio mediante redes neuronales se fundamenta en la investigación de nuevos algoritmos que cumplan con los requisitos de idoneidad que debe cumplir un modelo de red, tal y como se ha expuesto en el capítulo 3. Estos algoritmos se apoyan en nuevas formalizaciones matemáticas que amplían las posibilidades de la arquitectura «feedforward» en redes neuronales.

Antes de alcanzar este objetivo, en el capítulo 4 se exponen una serie de resultados matemáticos sobre redes neuronales con arquitectura «feedforward». Los resultados concretos de esta parte han sido:

1. Una demostración nueva de que una red neuronal «feedforward» con $n + 1$ neuronas en la capa oculta puede aprender $n + 1$ ejemplos con error cero.
2. La introducción de redes de interpolación aproximada («ai-nets») en el caso unidimensional. Estas redes no requieren entrenamiento y pueden interpolar aproximadamente un conjunto arbitrario de ejemplos distintos.
3. Una demostración constructiva de la convergencia uniforme de las ai-nets a cualquier función continua de una variable si la función de activación es sigmoideal. Una de las consecuencias es una demostración nueva del teorema de Cybenko.
4. Una prueba constructiva de que 2 neuronas pueden aproximar una función afín.

El resultado fundamental de este capítulo es el hallazgo de un algoritmo que garantiza un aproximante uniforme sin entrenamiento. El método de retropropagación del error no garantiza una adecuada aproximación, sólo interpola sin garantizar aproximación uniforme.

La acción sobre un objeto requiere utilizar distancias en el espacio en dos y tres dimensiones. En el capítulo 5 se formula un modelo autogenerado mediante árboles neuronales que puede realizar el cálculo de distancias, utilizando información del espacio. El resultado principal de esta parte es un algoritmo de generación de redes neuronales en forma de árbol, en dos pasos:

1. Aproximación afín por trozos por división adaptativa.

2. Reemplazamiento de los nodos no hoja y hoja del árbol de aproximación con redes de direccionamiento y computación.

En cada nodo hoja se sitúa una SLFN con dos neuronas en la capa oculta que aproxima localmente la función distancia. El procedimiento para construir la red se puede llamar de neuromorfogénesis, ya que se genera un árbol que se adapta a la función. La estructura de la red se determina por la función a aproximar.

Por último, se realizan una serie de experimentos computacionales para mostrar la eficiencia de aprendizaje del algoritmo propuesto:

1. Se analiza el comportamiento numérico de los árboles autogenerados en el aprendizaje de la función distancia a objetos en 2 y 3 dimensiones. Los resultados indican que dichos árboles generalizan adecuadamente y aprenden la función distancia a un objeto dado.
2. Se compara el rendimiento de los árboles generados con redes SLFN entrenadas por un método de regiones de confianza basado en el algoritmo de Newton (interior-reflexivo), un método de optimización muy eficiente.

Los resultados demuestran que el modelo es eficiente, en comparación con otros algoritmos (Ver sección 5.6). Además es biológicamente plausible y podría modelar funciones del córtex parietal de acuerdo con la necesidad de interactuar con los objetos, tal y como parecen sugerir investigaciones neurocientíficas acerca de la vía dorsal.

Bibliografía

Andersen R. A., Snyder L.H., Bradley D.C., y Xing J. (1997). Multimodal representation of space in the posterior parietal cortex and its use in planning movements. *Annual Review of Neuroscience* 20, 303-330.

Anthony M. (1994). Function learning from interpolation. London School of Economics, -Mathematics Preprint Series-73.

Anthony M. y Bartlett. P.L. (1999). *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge.

Astafiev S.V., Shulman G.L., Stanley C.M., Snyder A.Z., Van Essen D.C., y Corbetta M. (2003). Functional organization of human intraparietal and frontal cortex for attending, looking, and pointing. *Journal of Neuroscience* 23, 4689-4699.

Attali J.G. y Pagés G. (1997). Approximations of functions by a multilayer perceptron: a new approach. *Neural Networks* 10, 1069-1081.

Barhen J., Cogswell R., y Protopopescu V. (2000). Single iteration training algorithm for multilayer feedforward neural networks. *Neural Processing Letters* 11, 113-129.

Bartle R.G. (1976). *The Elements of Real Analysis*. John Wiley and Sons, New York.

Bechtel W. y Abrahamsen A. (1991). *Connectionism and the Mind: An Introduction to Parallel Processing in Networks*. Blackwell, Cambridge (MA), 1991.

Berntsen J. y Espelid T.O. (1991). Error estimation in automatic quadrature routines. *ACM Trans. Math. Software* 17, 233-252.

Binev P. y de Vore R. (2004). Fast computation in adaptive tree approximation. *Numer. Math.* 97, 193-217.

Binkofski F., Dohle C., Posse S., Stephan K.M., Hefter H., Seitz R.J., y Freund H.J. (1998). Human anterior intraparietal area subserves prehension. *Neurology* 50, 1253-1259.

Birman M.S. y Solomjak M.Z. (1967). Piecewise-polynomial approximations of functions of classes W_p^α . *Sb. Math.* 2, 295-317.

Blankenship J.E., Wachtel H., y Kandel E.R. (1971). Ionic mechanisms of excitatory, inhibitory and dual synaptic actions mediated by an identified interneuron in abdominal ganglion of *Aplysia*. *Journal of Neurophysiology* 34, 76-92.

Bliss T.V.P. y Lomo T. (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology* 232, 331-356.

Bliss A. y Su F.E. (2005). Lower bounds for simplicial covers and triangulation of cubes. *Discrete Comput. Geom.* 33, 669-686.

de Boor C. y Rice J.R. (1979). An adaptive algorithm for multivariate approximation giving optimal convergence rates. *J. Approx. Theory* 25, 337-359.

Cardaliaguet P. y Euvrard G. (1992). Approximation of a function and its derivative with a neural network. *Neural Networks* 5, 207-220.

Cavina-Pratesi C., Galletti C., Fattori P., Quinlan D.J., Goodale M., y Culham J. (2006). Event-related fMRI reveals a dissociation in the parietal lobes between transport and grip components in reach-to-grasp movements. *Society Neuroscience Abstracts* 32, 307-312

Chen K., Yang L., Yu X., y Chi H. (1997). A self-generating modular neural network architecture for supervised learning. *Neurocomputing* 16, 33-48.

Chen T., Chen H., y Liu R.-W. (1991). *A constructive proof and an Extension of Cybenko's approximation theorem*. In: Computing Science and Statistics: Proceedings of the 22nd Symposium on the Interface, Springer, Berlin, pp. 163-168.

Chui C.K., Li X., y Mhaskar H.N. (1994). Neural networks for localized approximation. *Mathematics of Computation* 63, 607-623.

Chui C.K., Li X., y Mhaskar H.N. (1996). Limitations of the approximation capabilities of neural networks with one hidden layer. *Advances in Computational Mathematics* 5, 233-243.

Cohen Y. E. y Andersen. R. A. (2002). A Common Reference Frame for Movement Plans in The Posterior Parietal Cortex. *Nature Reviews 3 (Neuroscience)*, 553-562.

Coleman T.F. y Li Y. (1994). On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. *Math. Progr.* 67, 189-224.

Coleman T.F. y Li Y. (1996). An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.* 6, 418-445.

Culham J.C. y Kanwisher N.G. (2001). Neuroimaging of cognitive functions in human parietal cortex. *Current Opinion in Neurobiology* 11, 157-163.

Culham J.C. (2004). *Human Brain Mapping Reveals a Parietal Area Specialized for Grasping*. En: Attention and Performance XX: Functional Brain Imaging of Visual Cognition. Oxford University Press, Oxford.

- Cybenko G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 2, 303-314.
- Dahmen W. (1982). Adaptive approximation by multivariate smooth splines. *J. Approx. Theory* 36, 119-140.
- Debao Ch. (1993). Degree of approximation by superpositions of a sigmoidal function. *Approx. Theory y its Appl.* 9, 17-28.
- DeVore R.A. y Yu X.M. (1990). Degree of adaptive approximation. *Math. Comp.* 55, 625-635.
- Duch W. y Jankowski N. (1999). Survey of neural transfer functions. *Neural Computing Surveys* 2, 163-212.
- Faillenot L., Toni L., Decety J., Grégoire M.C., y Jeannerod M. (1997). Visual pathways for object-oriented action and object identification. Functional anatomy with PET. *Cerebral Cortex* 7, 257-290.
- Faugier-Grimaud S., Frenois C., y Stein D.G. (1978). Effects of posterior parietal lesions on visually guided behavior in monkeys. *Neuropsychologia* 16, 151-168.
- Funahashi K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks* 2, 183-192.

Gallivan J.P., Cavina-Pratesi C., y Culham J. (2009). Is That within Reach? fMRI reveals that the Human Superior Parieto-Occipital Cortex Encodes Objects Reachable by the Hand. *The Journal of Neuroscience* 29 (14), 4381-4391.

Gallese V., Murata A., Kaseda M., Niki N., y Sakata H. (1994). Deficit of hand preshaping after muscimol injection in monkey parietal cortex. *Neuroreport* 5, 1525-1529.

Gamelin Th.W. y Greene R.E. (1999). *Introduction to topology*. Dover Publications, New York.

Gardner H. (1985). *The Mind's New Science*. Basic Books, New York.

Gaustschi W., y Inglese G. (1988). Lower bounds for the condition number of Vandermonde matrices. *Numer. Math.* 52, 241-250.

Goodale M.A., y Milner A.D.(1992). Separate visual pathways for perception and action. *Trends in Neuroscience* 15, 20-25.

Goodale M.A., Milner A.D., Jakobson L.S., y Carey D.P.(1991). A neurological dissociation between perceiving objects and grasping them. *Nature* 349, 154-156.

Grill-Spector K., Kourtzi Z., y Kanwisher N.(2001). The lateral occipital complex and its role in object recognition. *Vision Research* 41, 1409-1422.

Grundmann A. y Möller H.M. (1978). Invariant integration formulas for the n-simplex by combinatorial methods. *SIAM J. Numer. Anal.* 15, 282-290.

Haxby J.V., Grady C.L., Horwitz B., Ungerleider L.G., Mishkin M., Carson R.E., Herscovitch P., Shapiro M.B., y Rapoport S.I. (1991). Dissociation of object and spatial visual processing pathways in human extrastriate cortex. *Proc. Natl. Acad. Sci. USA* 88, 1621-1625.

Hebb D.O. (1949). *The Organization of Behavior*. Wiley, New York.

Hecht-Nielsen R. (1987). Counterpropagation networks. *Applied Optics* 26, 4979-4984.

Hecht-Nielsen R. (1987). *Kolmogorov's mapping neural network existence theorem*. Proceedings of the International conference on Neural Networks (San Diego), vol.III, 11-14, IEEE, New York.

Hinton G.E. y Anderson J.A. (editores). (1981). *Parallel Models of Associative Memory*. NJ.: Lawrence Erlbaum Associates, Hillsdale.

Hornik K., Stinchcombe M., y White H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359-366.

Hornik K., Stichcombe M., y White H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks* 3, 551-560.

Hornik. K. (1991) Aproximation capabilities of multilayer feedforward networks. *Neural Networks* 4, 251-257.

Huang G.-B. y Babri H.A. (1998). Feedforward neural networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks* 9-1, 224-229.

Huang G.-B.(2003). Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Trans. Neural Networks* 14, 274-281.

Huang G.-B., Zhu Q.-Y., y Siew C.-K. (2006). Extreme learning machine: Theory and applications, *Neurocomputing* 70, 489-501.

Huang G.-B., Zhu Q.-Y., y Siew C.-K.(2006). Real-time learning capability of neural networks. *IEEE Trans. Neural Networks* 17, 863-878.

Humphrey N.K., y Weiskrantz L. (1967). Vision in monkeys after removal of the striate cortex. *Nature* 215, 595-597.

Ito M. (1984). Neuronal Network Model. Capítulo 10, 115-130 en *The Cerebellum and Neural Control* Raven Press: New York.

Ito Y. (1996). Nonlinearity creates linear independence. *Advances in Computational Mathematics*, 189-203.

Ito Y. y Saito K. (1996). Superposition of linearly independent functions and finite mappings by neural networks. *Math. Scient.* 21, 27-33.

Ito T. (2001). Independence of unscaled basis functions and finite mappings by neural networks, *Math. Scient.* 26, 117-126.

Ito M. (2001). Cerebellar long-term depression: Characterization, signal transduction, and functional roles. *Physiological Review* 81 (3), 1143-1195.

Jacobs R.A., Jordan M.I., Nowlan S.J., y Hinton G.E. (1991). Adaptive Mixtures of local experts. *Neural Computation* 3, 79-87.

Jakobson L.S., Archibald Y.M., Carey D.P., y Goodale M.A.(1991). A kinematic analysis of reaching and grasping movements in a patient recovering from optic ataxia. *Neuropsychologia* 29, 803-809.

Jeannerod M. (1988). *The Neural and Behavioural Organization of Goal-Directed Movements*. Oxford University Press, Oxford.

Jeannerod M.(1994). The hand and the object: the role of posterior parietal cortex in forming motor representations. *Canadian Journal of Physiology and Pharmacology* 72, 535-541.

Jeannerod M., Arbib M.A., Rizzolatti G., y Sakata M. (1995). Grasping objects: the cortical mechanisms of visuomotor transformation. *Trends in Neuroscience* 18, 314-320.

Jong B.M. de, van der Graaf F.H., y Paans A.M. (2001). Brain activation related to the representations of external space and body scheme in visuomotor control. *Neuroimage 14*, 1128-1135.

Jordan M.I. y Jacobs R.A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation 6*, 181-214.

Karaivanov B., Petrushev P., y Sharpley R.C. (2003). Algorithms for nonlinear piecewise polynomial approximation: Theoretical aspects. *Trans. Amer. Math. Soc.* 355, 2585-2631.

Karnath H.O., y Perenin M.T. (2005). Cortical control of visually guided reaching:evidence from patients with optic ataxia, *Cerebral Cortex 15*, 1561-1569.

Kernighan B. W., y Ritchie D. M. (1988). *El lenguaje de Programación C*. Prentice-Hall, New York.

Koch Ch. (1999). *Biophysics of Computation*. Oxford University Press, Oxford.

Kress R. (1998). *Numerical Analysis*. Springer, New York.

Langley J.N. (1895). Note on regeneration of prae-ganglionic fibres of sympathetic. *Journal of Physiology 18*, 280-284.

Lascaux P. y Theodor T. (1986). *Analyse Numérique Matricielle appliqué a l'art de l'ingénieur*. Masson, Paris.

Leshno M., Lin V.Y., y Pinkus A. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks* 6, 861-867.

Lippman R.P. (1987). An introduction of computing with neural nets. *IEEE ASSP Magazine April*, 4-22.

Li X. (1996). Simultaneous approximation of multivariate functions and their derivatives by neural networks with one hidden layer. *Neurocomputing* 12, 327-343.

Li X. (2002). Interpolation by ridge polynomials and its application in neural networks. *Journal of Computational and Applied Mathematics* 144, 197-209.

Llanas B., Fernández de Sevilla M., y V. Feliú V. (2000). An iterative algorithm for finding a nearest pair of points in two convex subsets of R^n . *Comput. Math. Appl.* 40, 971-983.

Llanas B. (2005). Efficient computation of the Hausdorff distance between polytopes by exterior random covering. *Comput. Optim. Appl.* 30, 161-194.

Llanas B. y Sáinz F.J. (2006). Constructive approximate interpolation by neural networks. *J. Comput. Appl. Math.* 188, 283-308.

Malakooti y Yingqing Z. (1998). Approximating Polynomial Functions by Feedforward Artificial Neural Networks: Capacity Analysis and Design. *Applied Mathematics and Computation* 90, 27-52.

Marr D. (1982). *Vision*. Freeman, San Francisco.

Mhaskar H.N. y Michelli Ch.A. (1992). Approximation by superposition of sigmoidal and radial basis functions. *Advances in Applied Mathematics* 13, 350-373.

Mhaskar H.N. (1993). Neural networks for localized approximation of real functions. En: Neural Networks for Signal Processing: Proceedings of the 1993 IEEE-SP Workshop, 6-9 Sept 1993, IEEE, pp. 190-196.

Mhaskar H.N. (1993). Approximation properties of a multilayered feedforward artificial neural network, *Adv. Comput. Math.* 1, 61-80.

Milner A.D. y Goodale. M.A. (1995). *The Visual Brain in Action*. Oxford University Press, Oxford.

Minai A.A. y Williams R.D. (1993). On the derivatives of the sigmoid. *Neural Networks* 6, 845-853.

Mountcastle V.B, Lynch J.C., Georgopoulos A., Sakata H., y Acuna C. (1975). Posterior parietal association cortex of the monkey: command functions for operations within extrapersonal space. *Journal of Neurophysiology* 38, 871-908.

- Nakashima A. y Ogawa H. (2001). Noise suppression in training examples for improving generalization capability. *Neural Networks 14*, 459-469.
- Newell A. y Simon H.A. (1976). Computer science as empirical inquiry: Symbols and search, *Communications of the ACM 19*, 113-126.
- Nguyen M.H., Abbass H.A., y McKay R.I. (2006). A novel mixture of experts model based on cooperative coevolution, *Neurocomputing 70*, 155-163.
- Orden D. y Santos F. (2003). Asymptotically efficient triangulations of the d -cube. *Discrete Comput. Geom. 30*, 509-528.
- Pylyshyn Z.W. (1984). *Computation and Cognition*. MIT Press, Cambridge (MA).
- Pinkus A. (1999). Approximation theory of the MLP model in neural networks. *Acta Numerica*, 143-195.
- Ramón y Cajal S. (1890). Sur l'origine et les ramifications des fibres nerveuses de la moelle embryonnaire. *Anat. Anz. 5*, 609-613.
- Ramón y Cajal S. (1913). *Estudios sobre la Degeneración y Regeneración del Sistema Nervioso*. Imprenta de hijos de Nicolás Moya. Madrid.
- Redhead R. y Heywood M. (2005). *Evaluation of cluster combination functions for mixtures of experts*. en: Proceedings of the International Joint conference on Neural Networks, (IEEE, Montreal) 1154-1159.

Ritter G. (1999). Efficient estimation of neural weights by polynomial approximation. *IEEE Trans. Inform. Theory* 45, 1541-1550.

Rivlin Th.J. (1981). *An Introduction to the Approximation of Functions*, Dover, Mineola, New York.

Ronco E. y Gawthrop P. (1995). *Modular neural networks: A state of the art*. Technical Report: CSC-95026, Centre for Systems and Control. University of Glasgow, Glasgow, UK, 1995.

Rosenblatt F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 386-408.

Rumelhart D.E., Hinton G.E., y McClelland J.L. (1986) *Parallel Distributed Processing, volume 1: Foundations*. MIT Press, Cambridge (MA).

Rumelhart D.E., Hinton G.E., y Williams R.J. (1986). Learning representations by back-propagating errors, *Nature* 323, 533-536.

Sakata H., Taira M., Murata A., y Mine S. (1995). Neural mechanisms of visual guidance of hand action in the parietal cortex of the monkey. *Cerebral Cortex* 5, 429-438.

Sartori M.A. y Antsaklis P.J. A simple method to derive bounds on the size and to train multilayer neural networks. *IEEE Transactions on Neural Networks* 2-4, 467-471.

Schneider G.E. (1969). Two visual systems. Brain mechanisms for localization and discrimination are associated by tectal and cortical lesions. *Science* 163, 895-902.

Selfridge O.M. (1959). *Pandemonium: A paradigm for learning*. En D. V. Blake and A. M. Uttley, editors, Proceedings of the Symposium on Mechanisation of Thought Processes, London, 511-529.

Sharkey A.,(Ed).(1999). *Combining artificial neural nets. Ensemble and modular multi-nets systems*. Springer Verlag, London.

Shrivastava Y. y Dasgupta S. (1990). *Neural networks for exact matching of functions on a discrete domain*. En: Proceedings of the 29th IEEE Conference on Decision and Control, Honolulu, pp. 1719-1724.

Sontag E.D. (1992). Feedforward nets for interpolation and classification. *J. Comp. Syst. Sci.* 45, 20-48.

Sperry R.W. (1963). Chemoaffinity in the orderly growth of nerve fiber patterns and connections. *Proc. Natl. Acad. Sci. USA*, 50, 703-710.

Stein E.M. y Weiss G. (1975). *Introduction to Fourier Analysis on Euclidean Spaces*. Princeton University Press, Princeton.

Stichcombe M. y White H. (1989). *Universal approximations using feedforward networks with nonsigmoid hidden layer activation functions*. Proceedings of the International Joint Conference on Neural Networks (Washington), vol.I, pp.613-617. SOS printing. San Diego. California.

Szymanski P.T., Lemmon M., y Bett Ch.J. (1998). Hybrid interior point training of modular neural networks. *Neural Networks 11*, 215-234.

Taira M., Mine S., Georgopoulos A.P., Murata A., y Sakata H. (1990). Parietal cortex neurons of the monkey related to the visual guidance of hand movement. *Experimental Brain Research 83*, 29-36.

Tamura S. y Tateishi N. (1997). Capabilities of a four-layered feedforward neural network. *IEEE Transactions on Neural Networks 8-2*, 251-255.

Trevarthen C.B. (1968). Two mechanisms of vision in primates. *Psychol. Forsch. 31*, 299-337.

Tseng Y.H., Hwang J.N., y Sheehan F.H. (1997). Three-dimensional object representation and invariant recognition using continuous distance transform neural networks. *IEEE Trans. Neural Networks 8*, 141-147.

Ungerleider L.G. y Mishkin M. (1982). *Two cortical visual systems, En: Analysis of Visual Behavior*. MA: MIT Press, Cambridge, eds. D.J. Ingle, M.A. Goodale and R.J.W. Mansfield, 549-586.

Werbos P.J. (1995). *Backpropagation: Basics and new developments*. In Arbib, editor. The Handbook of Brain Theory and Neural Networks. MIT Press, Cambridge, MA, pages 134-139.

Widrow B. y Hoff. M.E.(1960). *Adaptive switching circuits*. En 1960 IRE WESCON Convention Record, (96-104). New York. IRE

Wiener N. (1948). *Cybernetics*. John Wiley Sons, New York, 1948.

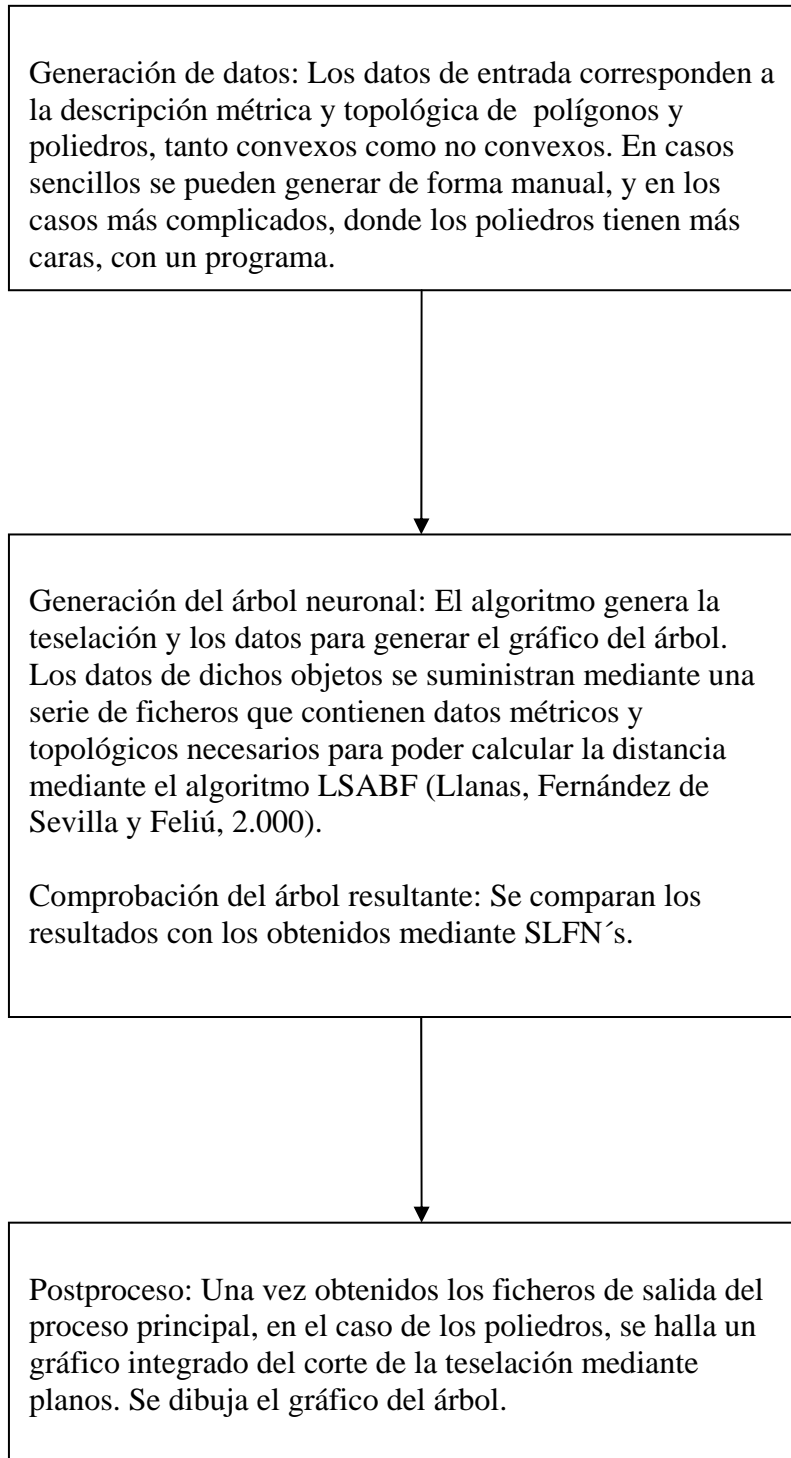
Walter J. y Ritter H. (1995). *Local PSOM and Chebyshev PSOMs improving the parametrized self organizing maps*. en: Fogelman-Soulie (ed.) International Conference on Artificial Neural Networks, Paris.

Xu Z.B. y Cao F.L. (2005). Simultaneous L^p -approximation order for neural networks. *Neural Networks* 18, 914-923.

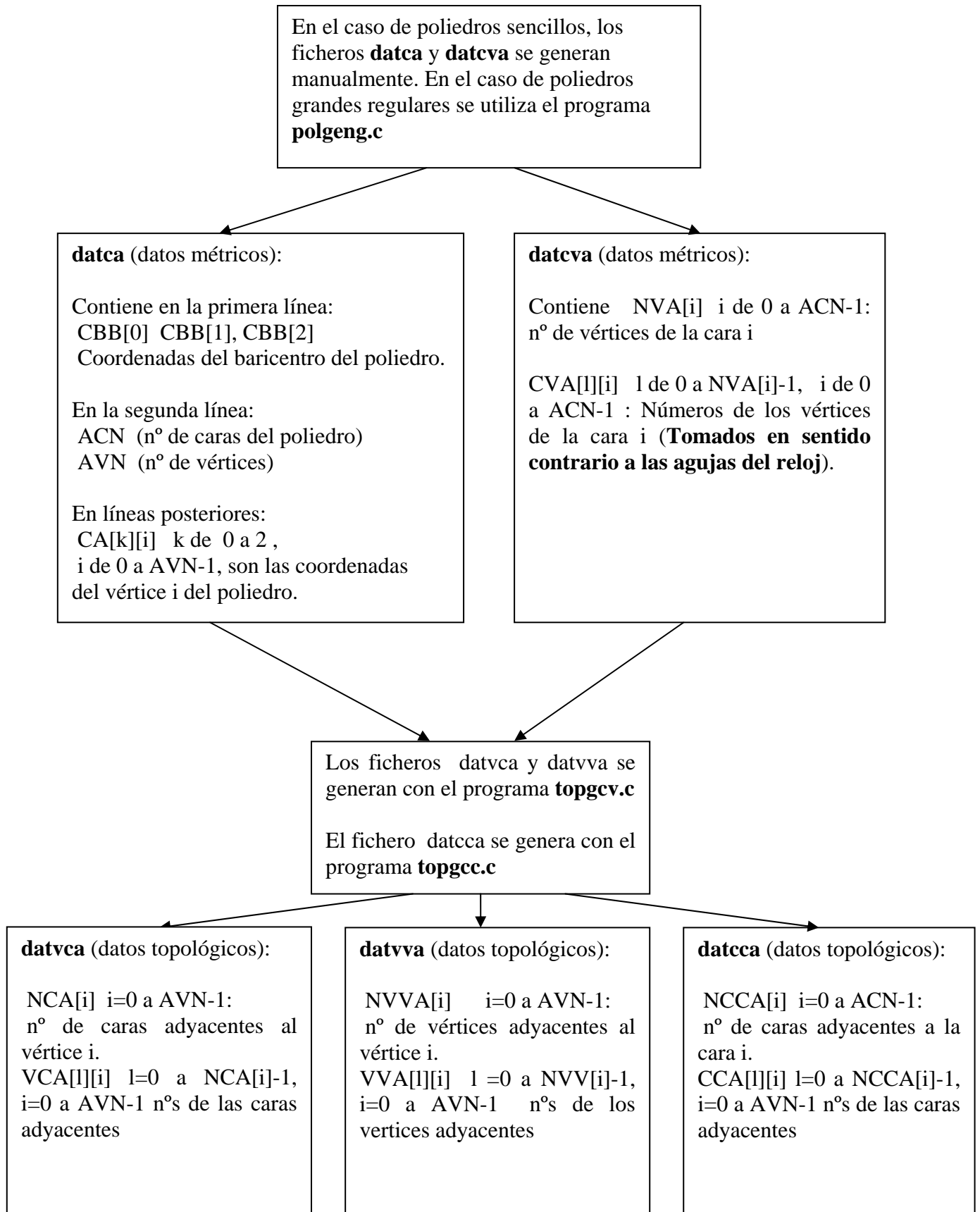
Yao X. (1999). *Evolving artificial neural networks*. Proceedings of the IEEE 87, 1423-1447.

Esquema General de los Programas

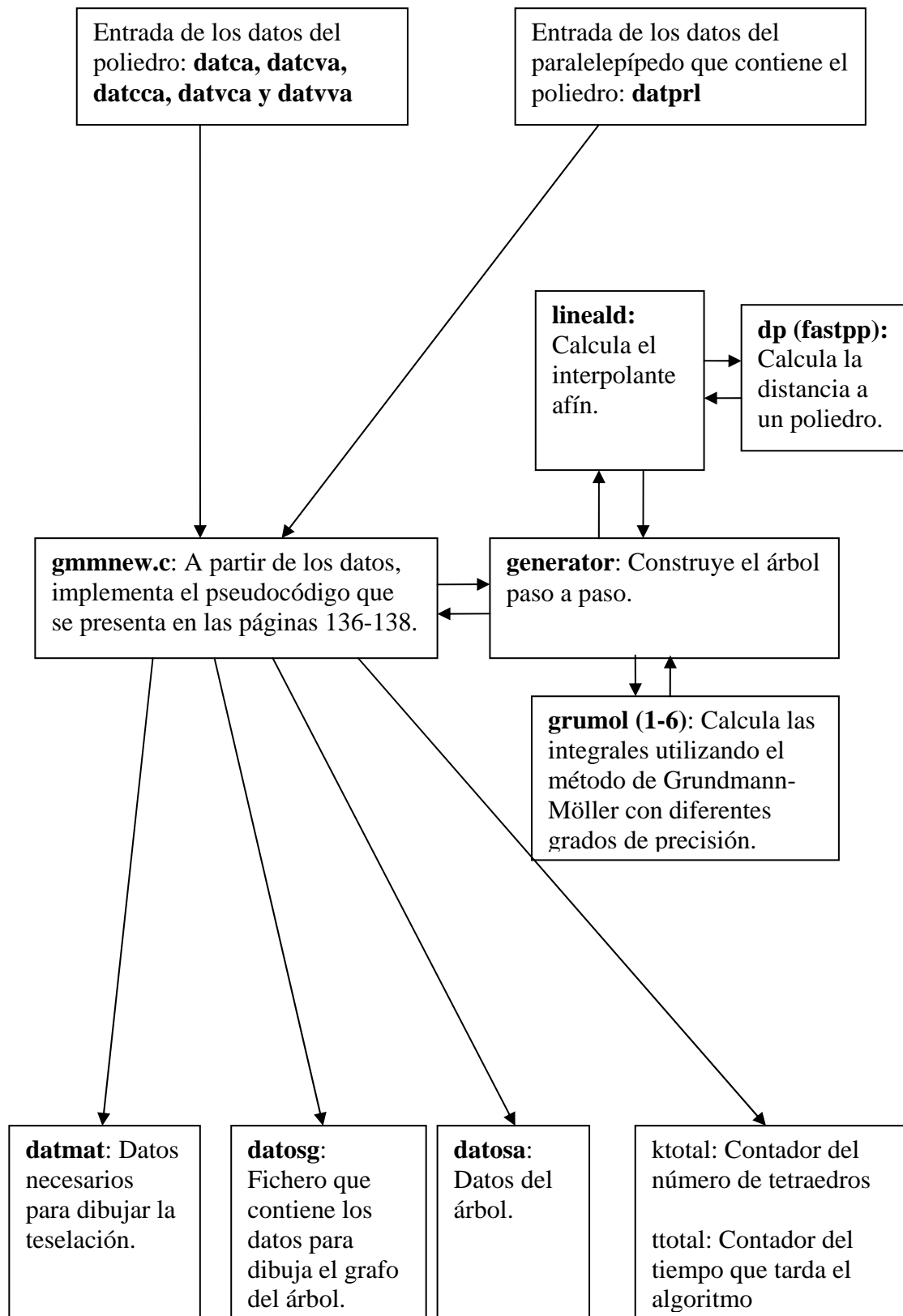
Esquema general de cálculo



Generación de datos de los poliedros

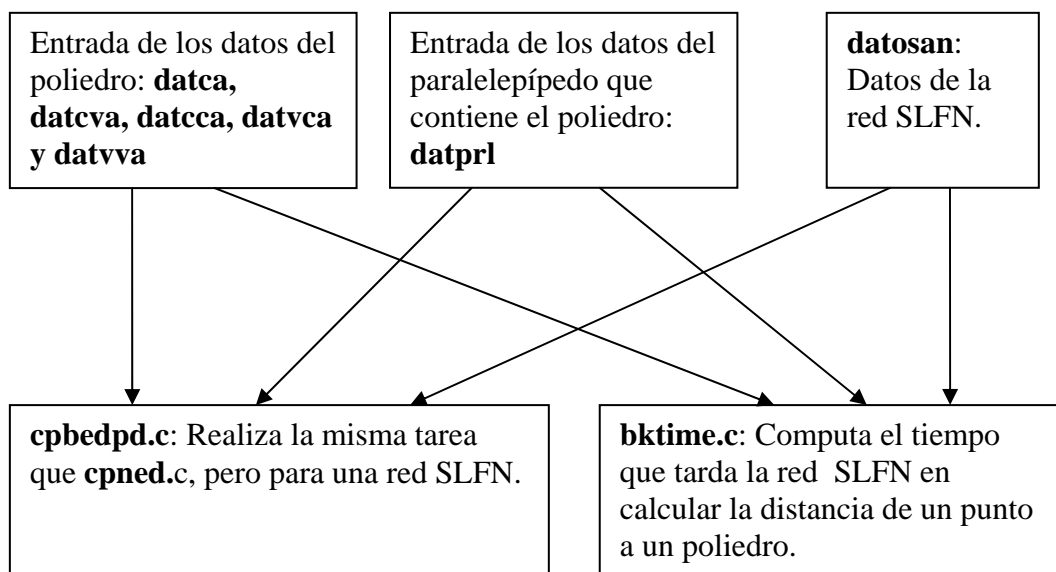
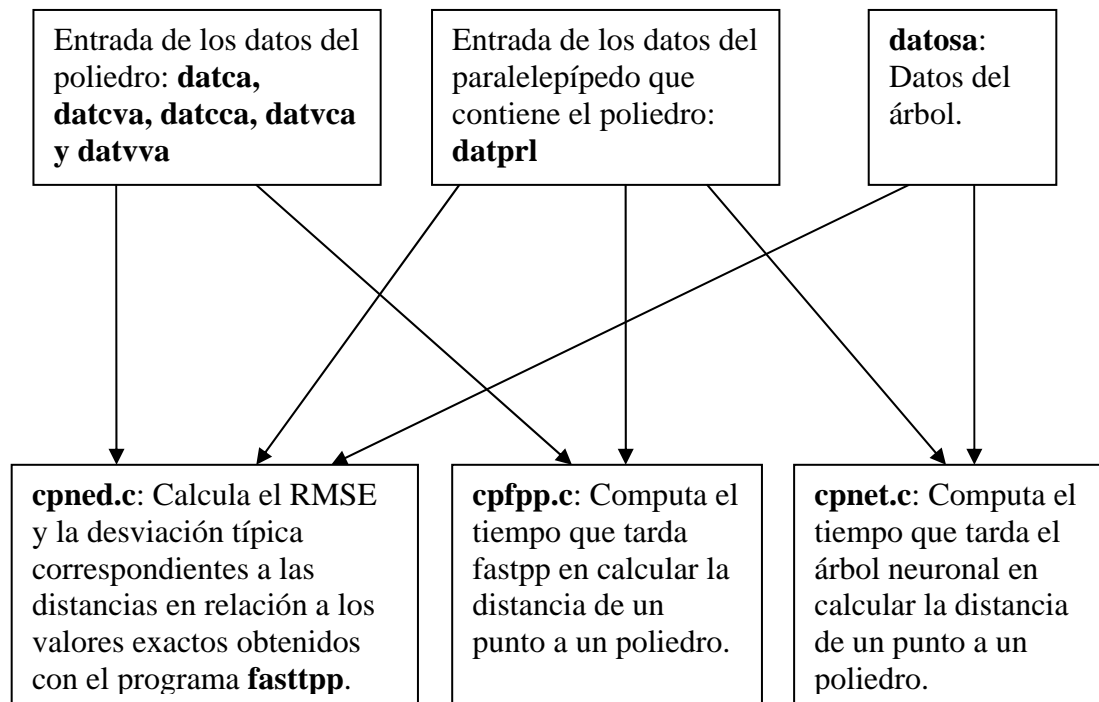


Generación del árbol neuronal



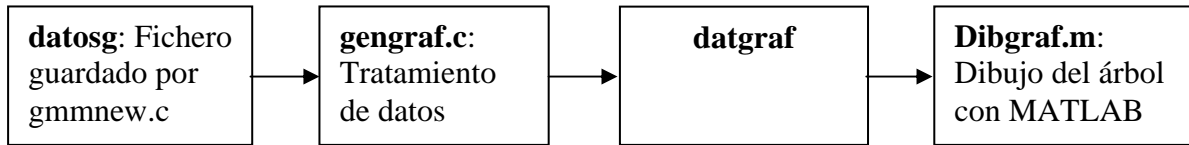
Efectividad del árbol neuronal. Comparación con redes SLFN

Además del tiempo de entrenamiento, se estudia el grado de aproximación de las distancias obtenidas utilizando el grado de aproximación de las distancias obtenidas utilizando el error cuadrático medio (RMSE). También se estudia el tiempo de cálculo que utiliza el árbol, una vez entrenado, para obtener la distancia de un punto a un poliedro (tiempo de recuerdo). Se hace una comparación con los mismos parámetros obtenidos a partir de redes SLFN.



Postproceso de poliedros

El postproceso ha consistido en la obtención de gráficos para visualizar los resultados. En primer lugar se ha realizado el dibujo del árbol neuronal mediante el siguiente proceso:



Para visualizar los resultados tridimensionales se han cortado las teselaciones 3D por planos perpendiculares al eje de las Z. Se ha calculado la intersección del plano con el poliedro que resulta en un polígono. Por otro lado se han calculado las intersecciones de los subtetraedros de la teselación con el plano. Los cálculos se han organizado de acuerdo al siguiente esquema:

